SESSION

NETWORK SECURITY I

Chair(s)

Prof. Kathy Liszka

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

A Secure Routing Scheme for Networks with Unknown or Dynamic Topology using A-star Algorithm

Joseph Flinn
Whitworth Univeristy
300 W. Hawthorne Rd.
Spokane, Washington 99251–2515
jflinn18@my.whitworth.edu

Hen Su Choi Ortiz
University of California San Diego
9500 Gilman Dr.
La Jolla, California 92093–0009
hechoi@ucsd.edu

Shengli Yuan
University of Houston-Downtown
One Main St.
Houston, TX 77002
yuans@uhd.edu

***Abstract*—In recent years, networks with unknown or dynamic topology are becoming more common. In such networks, each node only knows the network topology that it is connected to. Some examples of networks with these characteristics are Wireless Sensor Networks, Ad-Hoc Networks, and Vehicular Ad-Hoc Networks. The security of networks with unknown or dynamic topology is becoming a concern because of the increase in sensitive data that these networks are carrying. Therefore, the security of routing algorithms in networks with unknown or dynamic topology is becoming an area of active research.**

In this paper, we propose an application of the A-Star algorithm in order to enhance the security of routing in networks of unknown or changing topologies against attacks that rely on the predictability of the routing path chosen. We have modified the heuristic component of the A-Star algorithm to provide flexibility between security and efficiency. We investigated the effects of the heuristic component with various random factors, and compared the performance of our solution with the well-known Dijkstra's algorithm through computer simulations to give us a base to compare the efficiency of our algorithm. Our study confirmed that the proposed solution achieves a higher level of security and flexibility, however at the cost of taking the most efficient path.

***Keywords*—Secure Routing Algorithm, A-star Algorithm, A* Algorithm, Dynamic Network Topology, Unknown Network Topology, Network Security, Multipath routing**

I. Introduction

The security of *networks with unknown or dynamic topology* is very important. They are starting to be used in multiple areas of the industry. Wireless Sensor Networks (WSN) are being used in multiple situations to collect data [1] [2]. As the automobile industry seriously looks at the likelihood of self-driving cars, the security of the communication between cars is becoming a concern [3]. These networks have highly dynamic topology or unknown topology. This unknown or dynamic environment presents a problem in the routing of data in these networks.

In a Vehicular Ad-Hoc Network (VANET), part of the data that would be sent over the network would be the physical location of the vehicle. The location data could be used to route the vehicle on a different route if there is some event that will decrease the traffic flow. However, this data could present a major security problem [3]. If an attacker wants to harm a government official or another well-known person that uses VANETs, the attacker could gain access to the network and track the specific vehicle that their target uses. They would then know where the vehicle is and could coordinate an attack on the vehicle.

In this paper, we will be discussing use of the A-star algorithm applied as a secure path finding algorithm in these networks with unknown or dynamic topology. We will be presenting the A-star algorithm, discussing its strengths in these types of networks. Finally, we will present and discuss the comparison of the A-star algorithm to the well-known Dijkstras Algorithm through data that we have collected through simulation.

II. Literature Review

Networks with unknown or dynamic topology have become more widely used in a variety of applications. This type of environment means that each node only knows only a part of the network that it is connected to. The most prevalent examples are Wireless Ad-Hoc Networks (WANET), WSN, and VANETs. A WANET is a network that has no fixed infrastructure [4]. Thus a WANET is a network with a dynamic topology. Each node in the network has the ability to move its position in the network, leave the network, and come back to the network. The possible movement of the nodes leads to a network in which the included nodes do not know the position of other nodes in the network. Because of this mobility, it is difficult for conventional algorithms such as Open Shortest Path First (OSPF) to build and maintain a routing table because the physical location of the destination node may have changed.

OSPF is a very popular routing protocol because of the efficiency of the shortest path when used in routing [5]. Assuming there is no congestion, using the shortest path will get the packets to the destination the quickest. However, using this protocol makes the path chosen to take the data to the destination node very predictable because it will always take the shortest path. So OSPF is not very secure against attacks like sniffer attacks [6]. But OSPF wasn't designed to be secure, it was designed to return the shortest path.

WANETs do not have a base station or switch centers [4]. A level of trust is then required between the nodes to pass

the correct data and pass it correctly. But this trust presents a strong susceptibility to security attacks [7]. Compromised nodes and denial of service (DoS) attacks are easily produced because of the unattended nature of WANETs such as a WSN. These two attacks key attacks against WANETs [8]. There are other attacks such as: sniffers, sink hole (black hole), worm hole, selective forwarding, and Sybil attacks. Compromised nodes and sniffer attacks take advantage of the known routing algorithms that choose the shortest path. When the attacker knows the topology of the network, it is easy to find the placement of a sniffer or which node to compromise to compromise the data travelling between two specific nodes.

There have been many different proposals to overcome this security flaw [9] [7] [10] [11] [6]. Some of these proposals include cryptography [7] [1]. However, against a compromised node attack, cryptography is not a good security protocol all by itself. Once a node is compromised, the attacker may have access to the decryption and encryption keys that is used to secure the data [8].

Another solution that has been proposed is using multiple paths instead of just one [11] [6] [12]. This approach has proven through research to avoid sniffer, sink hole, and compromised node attacks. Compromised nodes forces traffic through themselves by making themselves more attractive to other nodes by emitting a low weight to send traffic through it. By using multiple paths, the algorithm won't always take the path that the compromised node wants the data to take. So the attacker has to work harder at getting all of the information that is sent from one router to another [6]. Instead of receiving all of the data with an accurately placed sniffer, the attacker will have to place multiple sniffers or compromise multiple nodes on the correct paths to intercept all of the data that is sent from one node to another.

There are multiple path search algorithms. One such algorithm is A-star Algorithm [9]. This algorithm is mostly used in robotics to plot the movement of robots from one point to another around obstacles. There has been a small amount of research into using the A-star algorithm in WANETs because of the unknown characteristics of these networks [9] [13]. In this algorithm each node only needs to know all of the nodes within one hop of itself. These nodes are also known as *neighbors*. Because A-star does not need to know the whole network, it can be applied to networks with dynamic or unknown topology [14]. Every node only needs to keep track of its neighbors changes instead of the whole network.

In past research, the A-star algorithm has been used as a search algorithm in WANETs. Dong and Li [9] presented research that used a version of the A-star algorithm for the search algorithm used for routing purposed in a WANET. Their proposal resolves the problem of when greedy forwarding fails. AlShawi et al. [13] proposed a new routing method that uses the Fuzzy approach and the A-star algorithm that extends the lifetime of the network. Their method balances the remaining power, traffic load, and minimum number of hops when considering the path used to transfer data. Li et al. [14] proposes an algorithm using greedy and A-star heuristic to minimize the hops and thus preserve the energy used in the system. Their algorithm adapts to unpredictable network topology. They focused this algorithm on the applications in home automation.

However, the above research did not focus on the security of these networks. They focused more on the search algorithm application and on the power efficiency of the networks. Other research regarding security for WANETs uses other algorithms that A-star. Shi and Li [10] present an algorithm that uses ant colony optimization to secure the data that is handled by the algorithm. Their algorithm prefers the next hop node that has the highest creditworthiness. Khiani et al. [11] propose a secure routing algorithm that utilizes multiple disjoint paths. Their technique also values the path with the highest remaining power since they are focusing on WSNs. Pagan et al. [6] uses a similar approach, but instead of valuing power, their research uses a random variable to choose the path. This contributes to the security of the network through randomness. Menaria et al. [15] presents an algorithm that identifies and removes compromised nodes from the plotted network. They are also focusing on the security and the efficiency of the network as a whole.

In this paper we are going to use the A-star algorithm with the main goal of security in a network with unknown or dynamic topology.

III. Solution

Using a path finding algorithm that finds the shortest path, or the path of least cost, is great when trying to optimize the path finding algorithm to find the fastest way to get data from the source node to the destination node. Dijkstra's Algorithm is a good example of this type of algorithm. Dijkstra's Algorithm has been the industry standard in routing protocol, via OSPF. This search algorithm finds the shortest path from one node to all of the other nodes in the graph. Dijkstra's Algorithm is run from each node in the network. Each node then populates an IP routing table with these shortest paths for future use. This is very helpful in a network with static topology. It only has to run the algorithm once initially and when it receives a network change update.

The problem is defined as follows: Given a network G = (V,E) where V is a set of nodes and E is a set of links between those nodes (consider them bidirectional). Also given is V_s and V_d where V_s is the source node and V_d is the destination node. Data must be transmitted from V_s to V_d. Traditionally, the shortest path has been valued because of the speed efficiency for transporting the data across the network. G is given to Dijkstra's Algorithm along with V_s and V_d. Dijkstra's algorithm then will return P where P is a shortest path between V_s and V_d. If Dijkstra's Algorithm is run multiple times, P will be the same assuming the network does not change.

There are two issues here: The first is a small issue. Algorithms that require the knowledge of G present a problem for networks with unknown or dynamic topology like WSNs and VANETs because the topology of the network cannot be easily known. The second issue is the security of the path chosen to transport the data between the nodes. If P is the shortest path every time, then P is very predictable. This predictability allows an attacker that knows part of the topology of the network to compromise certain nodes or the whole network using different attacks.

A-star is a search algorithm that does not need to know the entire network before it runs. However, it does assume that every node knows all of its neighboring nodes. Thus, it can be used natively in networks with unknown or dynamic topology. The algorithm starts with a source node and a destination node. As A-star is run, it starts from the source node and explores the network discovering nodes and adding them to a priority queue that is available to the source node. A-star utilizes a heuristic to prioritize the nodes to search first. As A-star starts to explore the network from the source node, it adds the cost to get to the neighbor to the value that the heuristic function returns. This value is the value that the priority queue uses to sort the nodes. It keeps searching all of the prioritized node's neighbors until it finds the destination node. Once it finds the goal node, the algorithm terminate and returns the path from the destination node to the source node.

Cryptography and multipath routing are the two main solutions that are being applied to secure data being routed. We chose to focus on multipath routing to minimize the overhead computation power required. Because of this overhead, we chose to pursue multipath routing and we decided to use the A-star algorithm as our search algorithm. For our algorithm, we are introducing a random factor that replaces the original A-star heuristic. Our heuristic returns a random value in a predetermined range. Since we are using a range of values for the random heuristic, the heuristic might overestimate the cost of reaching the destination (eg. Inadmissible Heuristic). However, the cost of reaching the destination node is only used to determine what node to explore next.

Our proposal will take only V_s and V_d. It assumes that all of the nodes in the graph knows all of their neighbors. Using HEURISTIC() as defined in Algorithm 1 (line 10), our algorithm will return a path P. P is guaranteed to be a path between V_s and V_d if V_d is in the network. P is not completely random because of how our algorithm rewards the total current cost of node when using HEURISTIC() (line 33 of Algorithm 1).

By using a random heuristic, we have made our path selection unpredictable. This unpredictability secures the data that is traveling along that path. As we will see from the simulations, using a smaller random range rewards a node's current total cost from the source node. If we keep increasing the random range, the difference in weights becomes less apparent and more randomness is added to the path selection.

The selection of the heuristic can be customized in function of the necessities of the network. By customizing the heuristic with a large random range, the path randomization is greater leading to a more secure routing algorithm. By using a smaller random range, you can increase the efficiency of the algorithm but at a cost to the security because the path randomization diminishes.

Another strength of our approach is that it can be adapted to automatically adjust the path when the network changes. This algorithm has to be run in set time intervals to keep up with changes in the dynamic network topology. Best case scenario, it can be used as a path discovery by using longer set time intervals and populate a routing table with this information. Worst case scenario is when the network is changing rapidly and it has to run our algorithm in shorter intervals. In an implementation of a routing protocol using our algorithm, the source node must be updated with the data from the node currently being explored. The data such as a current cost to get that node must be sent back to the source node so that its priority queue can be updated.

However, our algorithm does have a weakness. Our algorithm does not guarantee disjoint paths. Not using disjoint paths reduces the overall security of our algorithm against sniffer and spoofing attacks. By having multiple disjoint paths, there is a guaranteed data division. We are using different paths, but they may not be disjoint. So we do not have the guarantee of data division. But the path randomness compensates the weakness.

Algorithm 1 A* Algorithm with our Heuristic

```
1:  procedure GET_PATH(destination)
2:      add destination to path
3:      while destination.prev not NULL do
4:          destination = destination.prev
5:          path.append(destination)
6:          return path
7:      end while
8:  end procedure
9:
10: procedure HEURISTIC
11:     return random_int(0, upperbound)
12: end procedure
13:
14: procedure ROUTING ALGORITHM(source, destination)
15:     closedset
16:     openqueue
17:     add source to openqueue
18:
19:     while openqueue is not empty do
20:         currentNode = node in openqueue with lowest
    f_score
21:         if currentNode == destination then
22:             path = get_path(destination)
23:         end if
24:         add currentNode to closedset
25:         for each neighbor ∈ currentNode.neighbors do
26:             if neighbor in closedset then
27:                 continue
28:             end if
29:             tentative_g_score = g_score(currentNode) +
    distanceFrom(currentNode, neighbor)
30:             if neighbor not in openset or tentative_g_score
    < g_score(neighbor) then
31:                 neighbor.prev = currentNode
32:                 g_score(neighbor) = tentative_g_score
33:                 f_score(neighbor) = g_score(neighbor) +
    HEURISTIC()
34:             end if
35:         end for
36:     end while
37: end procedure
```

IV. Simulation

In this section we conduct simulations to determine the effectiveness of our algorithm. We used three graphs that were generated by LEDA programs: 10 node, 20 node and 40 node graphs with a nodal degree of 2.8 [16]. We did not have access to the software that created these graphs. We created a 60 node graph with the Networkx python library [17] matching the characteristics of the graphs created with the LEDA program. These characteristics are 2.8 total nodal degree and a random edge weight between 1 and 5. We tested our algorithm and compared it with Dijkstra's Algorithm shortest path results. We are using both weighted and unweighted versions of the generated graphs to do the testing. In the weighted graphs, the LEDA program assigned a random weight from between zero and five to each edge. In the unweighted graph, we are ignoring this weight and assigning all of the edges a weight of one.

In our simulation, we ran our algorithm for every combination of source and destination nodes. The simulation was run 1000 times and we took the average distance and average hop count to compare with the distance and hop count from when Dijkstra's Algorithm was run on the same graph. Since our algorithm could produce several different paths on different runs, we collected the frequency of time when the shortest path from the start node to the goal node was taken in order to evaluate the performance of our algorithm. We also wanted to test different ranges of the function HEURISTIC(). So we selected an upper bound wide enough in order to see the difference between ranges from zero to that upper bound. We were also interested in seeing how many paths were taken depending on this heuristic range and depending on the size of the graph. This number of paths would contribute to the security of the data. The more paths taken, the more diversified the data is and the harder it is to intercept all of it.

V. Results

Collecting the data from the simulation, we first compared the relationship between the heuristic and the frequency of the shortest path taken. This can be seen as efficiency versus security. As the heuristic increases, the number of paths used increases. This increase in number of paths used spreads the data traffic through the network making the data more secure. The shortest path can be seen as the most efficient path because it takes the shortest amount of time to travel, if the routing algorithm is optimized to the specific network.

The jump in the unweighted data can be explained with the evaluation using the heuristic. When the value of the heuristic upper bound is zero, the shortest path will always be taken. However, because the weight between all of the nodes in the unweighted graph is set to one, the path choice is more random because when the algorithm uses the path cost to weight the decision, the path costs are all the same. So the choice is made with only the random value from the heuristic.

We evaluated the affect the heuristic had on each type of graph: weighted and unweighted. For this evaluation, we used the distance from the source node to the destination node in the weighted graphs and we used the hop count in the unweighted graph. Using Mathematica [18], we found equations that relate the path cost (distance or hop count) with the range of heuristic

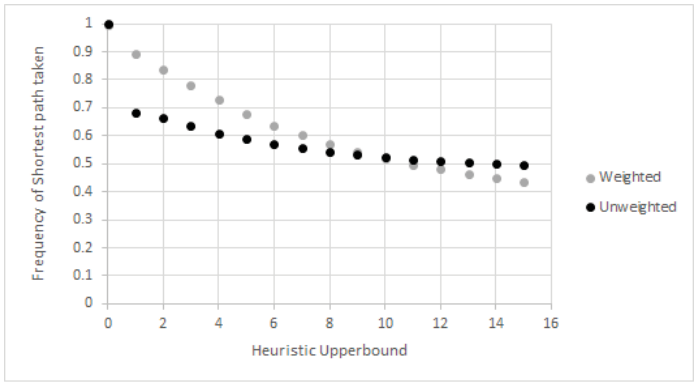


Fig. 1. Frequency of Shortest path taken with different Heuristic Ranges

and the size of the graph. Both of the resulting graphs are set up in the traditional x, y, z space where z is the height, x is axis to the left and y is to the right with the origin in the back corner. The axes are as follows: z is the distance taken (hop count for unweighted graphs), y is the heuristic upper bound, and x is the size of the graph.

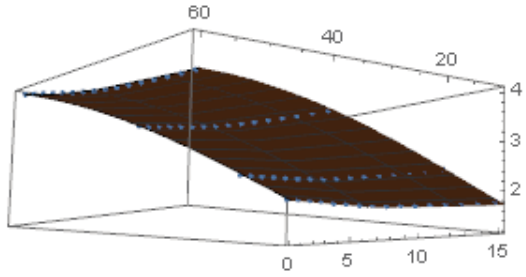


Fig. 2. Performance for unweighted graphs. Created with Mathematica [18]

$$z = 1.21 + 0.656e^{-0.126y} + 0.046x + (3.07 * 10^{-4})x^2 - (7.62 * 10^{-6})x^3 \quad (1)$$

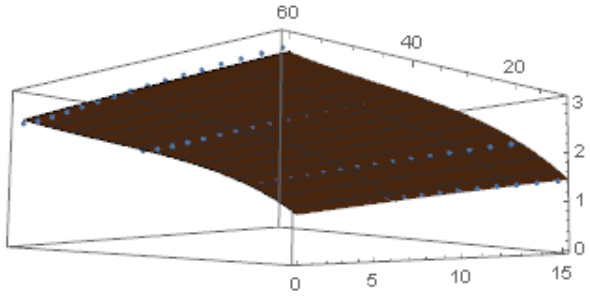


Fig. 3. Performance for unweighted graphs. Created with Mathematica [18]

$$z = 0.959 - 0.532e^{-0.046y} + 0.102x - 0.00196x^2 + 0.0000143x^3 \quad (2)$$

As z increases, the path taken gets longer. Thus as z increases, the path efficiency decreases. As the graphs show,

as the graph size gets larger, the performance of the path decreases. Surprisingly, the performance of the path sees an increase when looking at the weighted graph. This could be explained by supposing that there are multiple paths that differ only slightly in total cost. By adding a random heuristic, these paths would be taken more often increasing the path performance.

VI. Conclusion

As the use of MANETs, WSNs, and VANETs becomes more popular, the security of such networks becomes an issue. The conventional way of routing is to use the shortest path. However this creates a predictability of where the data is going to be. To address this, we proposed an algorithm that have a varying security and efficiency heuristic. This allows the same algorithm to be as secure or as efficient as the routing protocol calls for.

VII. Future Work

We plan to change the heuristic to give weight to hops, distance and a random variable. We also may choose to utilize other approaches (cryptography, secret handshakes, etc.) to make the algorithm even more secure. Another approach to increasing the security of this algorithm is increase the random value as the hop count increases. This would lead to a heuristic that places higher value on the shortest paths by hop count.

We also plan to test the algorithm on networks with different characteristics than the ones that we tested in this paper. We would like to see how our implementation of A* with our heuristic responds to extreme cases of network weighting; specifically when all of the weights are greater than the upper limit of the random range, or when all of the weights are less than the upper limit. We would be interested to see in what cases our changes improve and in what cases it doesn't improve.

We would also like to explore in a more detailed simulation, scenarios where there are compromised nodes that force network traffic through themselves. We want to know the effectiveness of our approach towards thwarting these types of attacks.

VIII. Acknowledgements

*The work presented in this paper is supported by NSF Grant 1262928

References

[1] P. L. R. Chze and K. S. Leong, "A secure multi-hop routing for iot communication," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp. 428–432, IEEE, 2014.

[2] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.

[3] A. Wasef and X. S. Shen, "Rep: location privacy for vanets using random encryption periods," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 172–185, 2010.

[4] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *Network, IEEE*, vol. 13, no. 6, pp. 24–30, 1999.

[5] G. Nakibly, A. Kirshon, D. Gonikman, and D. Boneh, "Persistent ospf attacks.," in *NDSS*, 2012.

[6] M. Pagan, A. Hession, and S. Yuan, "A security-enhanced routing algorithm with path randomization," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pp. 1137–1141, IEEE, 2015.

[7] A. Prabhu and R. Dhanaraj, "A flexible approach for securing manets," in *Recent Advances and Innovations in Engineering (ICRAIE), 2014*, pp. 1–4, IEEE, 2014.

[8] P. Manoj and S. S. Baba, "Random routing algorithms for wireless sensor networks," 2012.

[9] Z. Dong and M. Li, "A routing method of ad hoc networks based on a-star algorithm," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on*, vol. 2, pp. 623–626, IEEE, 2009.

[10] Q. Shi and Z. Li, "A secure qos routing algorithm based on aco for wireless sensor network," in *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*, pp. 1241–1245, IEEE, 2013.

[11] S. R. Khiani, C. Dethe, and V. Thakare, "Designing a secure and reliable node disjoint multipath routing algorithm and a survey on existing techniques," in *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, pp. 496–500, IEEE, 2014.

[12] A. S. Vijendran and J. V. Gripsy, "Enhanced secure multipath routing scheme in mobile adhoc and sensor networks," in *Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on*, pp. 210–215, IEEE, 2014.

[13] I. S. AlShawi, L. Yan, W. Pan, and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and a-star algorithm," *Sensors Journal, IEEE*, vol. 12, no. 10, pp. 3010–3018, 2012.

[14] X. H. Li, S. H. Hong, and K. Fang, "Wsnha-gahr: a greedy and a* heuristic routing algorithm for wireless sensor networks in home automation," *Communications, IET*, vol. 5, no. 13, pp. 1797–1805, 2011.

[15] V. K. Menaria, D. Soni, A. Nagaraju, and S. Jain, "Secure and energy efficient routing algorithm for wireless sensor networks," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, pp. 118–123, IEEE, 2014.

[16] "Algorithmic Solutions Software GmbH." http://www.algorithmic-solutions.com/leda/index.htm. Accessed: 2015-15-07.

[17] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, (Pasadena, CA USA), pp. 11–15, Aug. 2008.

[18] I. Wolfram Research, "Mathematica," 2015.

Baseline Operational Security Metrics for Industrial Control Systems

Guillermo A. Francia, III
Center for Information Security and Assurance, Jacksonville State University, USA

Abstract-- *Intrusions and attempted attacks on internet-facing industrial control systems have evolved into major national and international concerns as manifested by recent events. The security of these critical infrastructures is paramount to society's well-being. However, security procedures, standards, and policies must not only be in place but the practice of continuous improvement must also be exercised. To effectively implement such process, metrics must be developed. After all, improvement and measurement go hand in hand. In this paper, we present the development of operational security metrics that can be used to establish baseline security metrics for industrial control systems. We believe that these metrics can be utilized towards the realization of an enhanced security posture of our nation's critical infrastructures.*

Keywords: Security Metrics, Industrial Control Systems, SCADA, Threat Intelligence, Common Vulnerability Scoring System, NERC CIP

I. INTRODUCTION

The President's Commission on Critical Infrastructure Protection asserts that

> "Critical infrastructures are a network of independent, mostly privately-owned, man-made systems and processes that function collaboratively and synergistically to produce and distribute a continuous flow of essential goods and service" [1].

These include the nation's transportation, oil and gas production and storage, water supply, emergency services, government services, banking and finance, electric power, and telecommunications. Through the years, cyber security concerns on these systems continue to advance. The reliance of these systems has progressively evolved from operations that were not formerly computerized to computer based supervisory and control functions [2]. Compromising these computer based supervisory systems may lead to significant damage to these systems which we depend on and to catastrophic loss of human life [3].

Experience in securing traditional IT systems cannot simply be applied to industrial control systems (ICS) due to their differing requirements and development; a special set of security metrics is needed for industrial control systems.

One of the notable and prevalent security metrics is the Common Vulnerability Scoring System (CVSS) [4]. This open framework, which defines the characteristics and severity of software vulnerabilities, consists of three metric groups: Base, Temporal, and Environmental. The Base group covers the intrinsic qualities of vulnerability, the Temporal group reflects the properties of a vulnerability as it evolves over a time period, and the Environmental group represents the unique characteristics of a vulnerability in a specific system environment [4].

Stoddard, et al. [5] describe how security metrics are organized in three characterizations: organizational, operational, and technical. While organizational metrics assess the effectiveness of the organization's standards, polices, and procedures used to enhance security, operational metrics describe how these standards, policies and procedures are implemented on day-to-day functions. The technical metrics tend to provide a finer granularity with which measurements can be applied.

The North American Electric Reliability Corporation (NERC) created the Critical Infrastructure Protection (CIP) standards that cover the following: Critical Cyber Assets, Security Management Controls, Personnel & Training, Electronic Security, Physical Security, Systems Security Management, Incident Reporting and Response Planning, and Recovery Plans [6]. In addition, NERC publishes a guiding document [7] on violation severity level for non-compliance on each standard. These compliance metrics will be a major component in building the proposed ICS operational security metrics.

In a 2015 study published by Ponemon [8], 67% of respondents agree that the use of threat intelligence to improve security posture is cost effective. In the same study, 35 cyberattacks that eluded traditional defenses were uncovered by organizations that adopted threat intelligence to enhance their security postures. These and other

research data underscore the relevance of internal as well as external threat intelligence to cyber security. However, exactly how to ingest the intelligence and to successfully leverage it within an organization remains a big challenge [9]. Additional components of the proposed ICS operational security metrics will include the threat indicators that are gleaned from threat intelligence platforms.

The remainder of the paper is organized as follows: in section 2, we present a literature review of standards, risk assessment, and security metrics; the derivation of operational security metrics for ICS is described in section 3; section 4 covers operational security visualization; finally, we offer concluding remarks and provide a roadmap to future research directions in the last two sections.

II. BACKGROUND

The increasing network connectivity witnessed in Supervisory Control and Data Acquisition (SCADA) systems raises cyber security concerns for Industrial Control Systems (ICS) facilities [10]. Further, the various and numerous instrumentation and control systems, mingled with external offices and corporate business systems around it, creates heterogeneous environment that is difficult to monitor and maintain against cyber attacks [10].

Threats to industrial control systems (ICS) include "adversarial sources such as hostile governments, terrorist groups, industrial spies, disgruntled employees, malicious intruders, and natural sources such as from system complexities, human errors and accidents, equipment failures and natural disasters," [11]. A comprehensive list of threats to ICS is described in detail by Stouffer, Falco, & Scarfone [11]. Further exacerbating the current state of ICS security is the existence of vulnerabilities specific to the peculiarities of these systems as documented in published works found in [3], [12], [13] and [14].

Standards

Standards and best practices have been developed to promote the safety and security of control systems. The Common Criteria for Information Technology Security Evaluation (ISO 15408:2012) [15] provides the general evaluation methodology of Information Technology products. The Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (IEC 615908:2005) [16] provides a framework towards achieving functional safety in electrical, electronic or programmable electronic systems. The North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) Standards [6] provide cyber security framework for the protection of Critical Cyber Assets needed to manage Bulk Electric System reliability. The Department of Homeland Security (DHS) Control Systems Security Program (CSSP) has developed a control systems cyber security framework [17], consisting of seven dimensions of cyber security for control systems, to assist operators of critical infrastructure in managing their respective security posture.

Risk Assessment

Security risk management involves the identification, analysis, treatment, and monitoring of risk. Risk assessment is central to all of these processes. The Risk-to-Mission Assessment Process (RiskMAP) [18] is a risk assessment methodology designed for a typical Process Control System (PCS). This self-documenting methodology facilitates a better understanding among management the relevance of technical risk on possibly enabling adverse effects on business processes.

Houmb, et al. [19] argue that the constraint presented by the exposure of company confidential data makes quantitatively estimating risk a challenge. Faced with this predicament, they developed a risk estimation model based on publicly available data, the Common Vulnerability Scoring Systems (CVSS) [4]. The CVSS Risk Level Estimation Model [19] calculates the security risk level based on the vulnerability information and impact estimate derived from the CVSS.

The Carnegie Mellon University's CERT Coordination Center developed the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [20], which is a suite of tools, techniques, and methods for risk-based information security strategic assessment and planning. It uses the event/fault tree model to analyze threats to critical assets.

The interested reader is referred to [21] for a comprehensive review of security best practices and risk assessment of SCADA and ICS.

Security Metrics

A security metric is viewed as a key for an organization in fulfilling its responsibility to manage and secure the information system [22]. A good metric should measure the relevant data that satisfy the needs of decision makers and should be quantitatively measurable, accurate, validated on a solid base, inexpensive to execute, able to be verified independently, repeatable, and scalable to a larger scale [23]. Chew, et al. [24] developed 19 candidate security metrics that can be further developed to better fit and align with the level of implementation and scope of a particular organization's security

program. Francia and Jarupathirun [25] describe the development of Security Key Performance Indicators (SKPIs) following the guidelines advocated by Eckerson [26].

The metrics proposed by the Department of Homeland Security (DHS) are measured on each stage of operation in an industrial control system [17]. In essence, DHS proposed the division of security metrics into four different stages of industrial control system operations, because each stage of an industrial control system has several unique requirements. These stages are implementation, day-to-day operations, maintenance, and disaster recovery.

The implementation stage metrics cover the creation of security documentation, the design of security policies, and the establishment of plans for security awareness and training for operators.

During the operational stage, the metrics that are defined mostly include measurements of vulnerability that pertains to data exfiltration, system misconfiguration, authentication failures, etc. Also included in his stage are the execution of the security awareness and training plans, the implementation of incident response plans, and the execution of security audits, among others.

In the maintenance stage, the metrics may include the number of rogue change days, adequacy of test facilities, number of days to restore the system to the correct configuration, and the adequacy of patch testing which ensures that the ICS is returned to normal operations without introducing additional vulnerabilities due to incorrect configurations [10].

Finally, the disaster recovery stage describes the industrial control system during an unscheduled or unintentional shut down due to a cyber attack, a misstep in configuration, a natural disaster, or a loss of a supporting component. The metrics in this stage include restoration time, adequacy of back-up components, and other recovery measures [10].

In this paper, we focus our attention on the development of operational security metrics of industrial control systems. We extend the set of metrics on operational security for ICS that were first proposed in [10].

III. OPERATIONAL ICS SECURITY METRICS

The mapping of metrics towards operational stages and metric categories enables the analysis of the security posture of the industrial control system to determine which areas of security are strong and which areas need improvement. For instance, the use of operational stages also determines how the industrial control system will fair when trying to recover from a disaster. If low values occur for operational metrics, it can indicate to the organization that implementation of organization policies are not being adequately fulfilled. The challenge is in the development of such metrics.

CVSS Score

CVSS, an open framework characterizing the vulnerabilities of software and systems, consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic values of a vulnerability; the Temporal group represents the vulnerability property as it evolves over a time period; the Environmental group represents the characteristics of vulnerability that are unique to a specific environment [4]. The last group of vulnerability metrics lends itself perfectly with that of ICS. An actual vulnerability scan on an ICS server yields a CVSS score of 5.7 and a vector:

[AV:A/AC:L/PR:H/UI:R/S:C/C:N/I:H/A:N]

which translates to an Access Vector (AV) of Adjacent Network (A), a Low (L) Attack Complexity (AC), with High (H) Privileges (PR), a Required(R) User Interaction(UI), a Changed (C) Scope (S), with No (N) Confidentiality Impact, a High (H) Integrity Impact, and No (N) Availability Impact(A). Using this baseline score and further augmenting it with the Temporal group score and the Environmental group score, the final CVSS score is 7.6 with vector:

[AV:A/AC:L/PR:H/UI:R/S:C/C:N/I:H/A:N/E:P/RL:T/RC:R/CR:X/IR:X/AR:X/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:N/MI:H/MA:H]

An explanation about the Environmental section of the CVSS vector is in order. The Modified Attack Vector (MAV) may come from the Network (N), the Modified Access Complexity (MAC) is Low (L), the Modified Privileges Required (MPR) is Low, it requires ® a User Interaction (MUI), and the Modified Scope (MS) is Changed (C), i.e. it could affect resources beyond the authorization privileges enforced by the component. A summary of each group score is depicted in Figure 1.

NERC CIP Score

The NERC CIP score is derived from the published NER CIP Violation Severity Level (VSL) [7] Matrix. The violation severity levels are rated as Low (0-2), Moderate (3-5), High (6-8), and Severe (9-10). Each level is given a numeric range so as to remain within the range of the CVSS score.

In contrast to the CVSS scoring system where each component is assessed with a vulnerability measure, the NERC CIP score covers the entire organization. For instance, a scoring system that

covers CIP-007-5-R3 which requires the implementation of documented processes for Malicious Code Prevention will include the following metrics:

- (0-2:Low) One or more documented processes were implemented but the testing of signatures or patterns did not assess or correct deficiencies.
- (3-5:Moderate) One or more documented processes were implemented but the mitigation of malicious threat and the assessment or correction of deficiencies was not applied.
- (6-8:High) One or more documented processes were implemented but failed to deploy methods to deter, detect or prevent malicious code and have identified deficiencies but did not assess or correct those.
- (9-10:Severe) No documented processes were implemented. Deficiencies were identified but not assessed nor corrected.

Threat Intelligence Score

The Threat intelligence (TI) data is obtained from open-source or commercial treat intelligence exchanges. In Table 1, we delineate the scoring classifications according to the categories that are provided by Mateski, et al. [27] . Using the TI scoring guide the score is calculated using the following:

$$TI\_Score = \sum_{i=1}^{n} TC_i / n$$

A sample worksheet on a hypothetical threat intelligence report is shown in the following:

Incident:	Data Exfiltration	(10)
Target System:	ABB-PLC Identical	(10)
Timeline:	1 Month	(6)
Covert Activity:	Network Monitoring	(5)
Attack Vector:	Phishing	(2)
Attack Sophistication:	Advanced	(8)
Anti-virus Signature:	None	(8)
Physical Interaction:	None	(0)
Obfuscation:	None	(0)
Data Compromise:	Full	(10)
Attribution:	Known bad actor	(10)

TI_Score = 6.27

While the CVSS metrics apply to individual system components and the NERC CIP score appraises the security posture of an entire organization, the TI Score assesses the risk confronting an entire system infrastructure.

Aggregated Operational Security Metric

The Operational Security Metric (OSM) is an aggregation of the three metrics that are previously defined. However, purely taking the mean value of all the CVSS scores of system components is inadvisable due to the possibility of data being skewed at the low or high end of the spectrum. Thus, we utilize the median value as the representative value for the CVSS score. Thus,

OSM = median(CVSS_Score) + NERC_CIPScore + TI_Score

The upper bound for this metric is 30, indicating the riskiest security posture, and the lower bound is 0, indicating an extremely well protected entity.

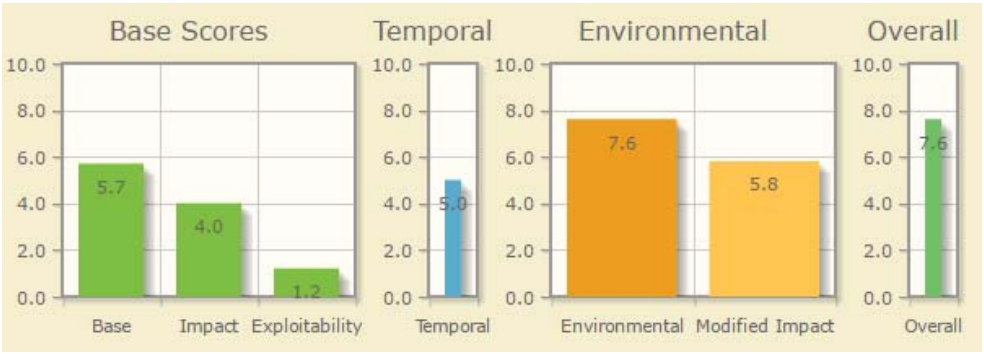


Figure 1. The Vulnerability Group Score as Calculated by the CVSS 3.0 Calculator [28]

Table 1. ICS Threat Category Scoring

THREAT CATEGORY	SCORING GUIDE SPECIFIC to ICS
Incident (IN)	Web, Recon, Probe (0-1); Denial of Service (2-5); Unauthorized Access, Remote Access, Data Exfiltration, C & C (6-10)
Target System (TS)	Barely related (0-3); Semi Related (4-6); Closely Related (7-8); Identical (9-10)
Timeline (TI)	Day (10); Week(7-9); Month (4-6); Year(0-3)
Covert Activity (CA)	User or Administrator (0-2); Network Monitoring (3-6); Anti-virus, Intrusion Detection Systems, Event Logs, File Signature (7-10);
Attack Vector (AV)	Phishing and Social Engineering (0-3), Malicious Software(4-5); Remote Access (6-7); Insider Access (8-10)
Attack Sophistication (AS)	Kiddie Script (0-2); Moderate (3-5); Advanced (6-10)
Anti-Virus Signature (AV)	Existing (0-3); Unknown (4-7); Non-existing (8-10)
Physical Interaction (PI)	No Access (0-2); Limited Access (3-6); Full Access (7-10)
Obfuscation (OB)	No Obfuscation (0-2); Limited Obfuscation (3-6); Totally Obfuscated (7-10)
Data Compromise (DC)	None (0-2); Limited (3-6); Total Exfiltration (7-10)
Attribution (AT)	Unknown source (0-2); Limited Knowledge(3-6); Known Bad Actor (7-10)

IV. SECURITY METRIC VISUALIZATION

Security visualization presents key security metrics in a form that are easily recognized, analyzed and interpreted for possible mitigation and/or correction. Seminal research works in the visualization field can be found in [29], [30], and [31]. In order to guide the ICS security professional in using the proposed operational security metrics, we developed proof-of-concept performance dashboards components for some of the metrics described above. These visual tools are depicted on Figures 1, 2, and 3.

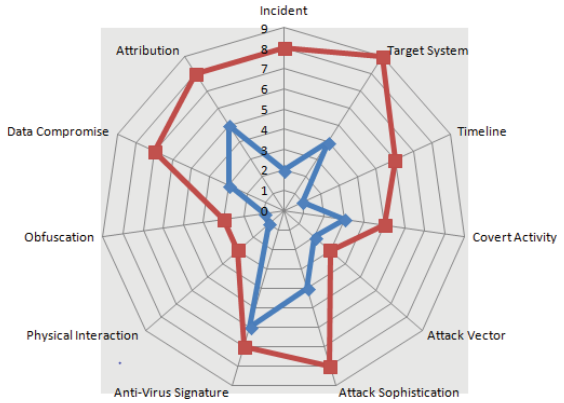


Figure 2. Threat Intelligence Metrics Chart

V. CONCLUSION

In this paper, we presented a compelling motivation for the need to protect our critical infrastructures. We provided a literature review of standards, risk assessment tools and techniques, and security metrics leading to the development of the proposed ICS metrics. In addition, we made a convincing argument to support the significance of security metrics with special emphasis on ICS security.

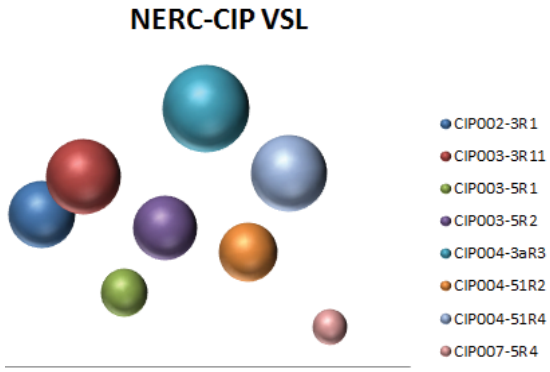


Figure 3. NERC-CIP Security Metrics

We developed a set of security metrics for ICS that are proposed to address the various and most critical issues confronting these systems by aggregating three sets of metrics that provide a comprehensive snapshot of the system's security posture. The most novel feature of this new set of metrics is in its ability to provide situational security awareness through continuous monitoring. We feel that this contribution to the advancement and understanding of security and protection of vital components of critical infrastructures will stimulate additional research and study in this area of national and global need.

An added feature of this research study is that on security metric visualization. We provided proof-

of-concept dashboard components to assist intelligent and interactive visual analytics on data metrics. We shall continue our efforts on developing such system and analyze its effectiveness.

VI. FUTURE RESEARCH DIRECTIONS

The proposed set of metrics introduced in this paper requires an elaborate validation. Thus, we propose the following future research directions:

- Design and implement simulation models that can validate the effectiveness of the proposed security metrics;
- Investigate the adaptability of the set of metrics to various industrial settings;
- Develop operational key security performance indicators for a typical industrial environment; and
- Design and implement a real-time continuous monitoring system that can provide visual analytics of security metrics.

VII. ACKNOWLEDGEMENTS

This work is supported in part by a Center for Academic Excellence (CAE) Cyber Security Research Program grant (Grant Award Number H98230-15-1-0270) from the National Security Agency (NSA). Opinions expressed are those of the author and not necessarily of the granting agency.

VIII. REFERENCES

[1] R. T. Marsh, et al. "Critical Foundations, Protecting America's Infrastructures, The Report of the President's Commission on Critical Infrastructure Protection.," President's Commission on Critical Infrastructure Protection, Washington, DC, 1997.

[2] D. Kroll, Securing Our Water Supply, Tulsa: PennWell, 2006.

[3] R. Krutz, Securing Scada Systems, Indianapolis: Wiley, 2006.

[4] Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System v3.0," 2016. [Online]. Available: https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf. [Accessed 22 March 2016].

[5] M. Stoddard, D. Bodeau, R. Carlson, C. Glantz, Y. Haimes, C. Lian, J. Santos and J. Shaw, "Process Control System Security Metrics–State of Practice," *I3P Institute for Information Infrastructure Protection Research Report,* 2005.

[6] North American Electric Reliability Corporation (NERC), "Critical Infrastructure Protection (CIP) Standards," North American Electric Reliability Corporation (NERC), 2015.

[7] NERC, "Complete Violation Severity Levels Matrix Encompassing All Commission-Approved Reliability Standards.," 29 March 2016. [Online]. Available: http://www.nerc.com/pa/stand/vsl matrix/vsl_matrix_complete_2016_03_29.docx. [Accessed 01 April 2016].

[8] L. Ponemon Institute, "The Importance of Cyber Threat Intelligence to a Strong Security Posture," March 2015. [Online]. Available: http://www.brightcloud.com/pdf/CyberThreatIntelligenceReport2015.pdf. [Accessed 2016].

[9] Lockheed Martin Corporation, "Seven Ways to Apply the Cyber Kill Chain with Threat Intelligenc Platform," 2015. [Online]. Available: http://informationsecurity.report/Resources/Whitepapers/80a2f446-b9a8-47c0-804c-7bd35d7f6a51_Seven%20Ways%20to%20Apply%20the%20Cyber%20Kill%20Chain%20with%20a%20Threat%20Intelligence%20Platform.PDF.

[10] G. A. Francia and X. P. Francia, "Critical Infrastructure Protection and Security Benchmarks," in *Encyclopedia of Information Science and Technology, Third Edition*, Hershey, PA, IGI Global, 2015, pp. 4267-4278.

[11] K. Stouffer, J. Falco and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publication,* 2008.

[12] S. C. Patel and Y. Yu, "Analysis of SCADA Security Models," *International Management Review,* 2007.

[13] E. J. Byres, M. Franz and D. Miller, "The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems," *International Infrastructure Survivability Workshop (IISW'04), IEEE,* 2004.

[14] V. M. Igure, Security Assessment of SCADA Protocols, VDM Verlag, 2008.

[15] I. 15408, "Common Criteria for Information Technology Security Evaluation," 2012. [Online]. Available: https://www.niap-ccevs.org/Documents_and_Guidance/cc_docs/CCPART1V3.1R4.pdf. [Accessed 22 March 2016].

[16] I. E. C. (IEC), "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems," 20 January 2005. [Online]. Available: https://webstore.iec.ch/publication/5514. [Accessed 23 March 2016].

[17] Department of Homeland Security, "Primer Control Systems Cyber Security Framework and Technical Metrics," June 2009. [Online]. Available: http://ics-cert.us-cert.gov/sites/default/files/documents/Metrics_Primer_7-13-09_FINAL.pdf.

[18] P. Kertzner, J. Watters, D. Bodeau and A. Hahn, "Process Control System Security Technical Risk Assessment," 2008.

[19] S. H. Houmb, V. N. Franqueira and E. A. Engum, "Quantifying security risk level from CVSS estimates of frequency and impact," *Journal of Systems and Software,* vol. 83, pp. 1622-1634, 2010.

[20] C. Alberts and A. Dorofee, Managing Information Security Risks; The OCTAVE Approach, Addison-Wesley Professional Publishing, 2003.

[21] G. A. Francia, D. Thornton and J. Dawson, "Security Best Practices and Risk Assessment of SCADA and Industrial Control Systems," in *2012 Internationation Conference on Security and Management (SAM'12)*, Las Vegas, NV, 2012.

[22] S. C. Payne, "A Guide to Security Metrics," 19 June 2006. [Online]. Available: http://www.sans.org/reading-room/whitepapers/auditing/guide-security-metrics-55. [Accessed 06 November 2013].

[23] O. S. Saydjari, "Is Risk a Good Security Metric?," in *Proceedings of the 2nd ACM Workshop On Quality of Protection (QoP'06)*, New York, NY, 2006.

[24] E. Chew, M. Swanson, K. Stine, N. Bartol, A. Brown and W. Robinson, "NIST Special Publication 800-55 Performance Measurement Guide for Information Security," July 2008. [Online]. [Accessed 06 November 2013].

[25] G. A. Francia and S. Jarupathirun, "Security Metrics-Review and Research Directions," in *International Conference on Security and Management (SAM'09)*, Las Vegas, NV, 2009.

[26] W. Eckerson, Performance Dashboards: Measuring, Monitoring, and Managing Your Business, 2nd ed., Hoboken, NJ: John Wiley & Sons, Inc., 2011.

[27] M. Mateski, C. M. Trevino, C. K. Veitch, J. Michalski, J. M. Harris, S. Maruoka and J. Frye, "Cyber Threat Metrics," Sandia National Laboratories, Albuquerque, NM, 2012.

[28] "Common Vulnerability Scoring System Version3 Calculator," National Vulnerability Database (NVD), 5 April 2016. [Online]. Available: https://nvd.nist.gov/CVSS/v3-calculator. [Accessed 5 April 2016].

[29] E. R. Tufte, Visual Explanations: Images and Quantities, Evidence and Narrative, Graphics Press, 1997.

[30] S. K. Card, J. D. Mackinlay and B. Shneiderman, Eds., Readings in Information Visualization: Using Vision to Think (Interactive Technologies), San Diego, CA: Academic Press, 1999.

[31] E. R. Tufte, The Visual Display of Quantitative Information, 2nd ed., Graphics Press, 2001.

Anomaly Detection in Smart Grid using Wavelet Transform and Artificial Neural Network

Maryam Ghanbari, Ken Ferens and Witold Kinsner
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada
{ghanbarm@myumanitoba.ca, Ken.Ferens@umanitoba.ca, Witold.Kinsner@umanitoba.ca}

Abstract— *This paper presents a scheme for detecting anomalous power consumption patterns attack using wavelet transform and artificial neural network for smart grid. The main procedure of the proposed algorithm consists of following steps: I) Creating normal and anomaly patterns of power consumption to train the proposed method. II) Wavelet transform is applied on power consumption patterns to extract features. III) Training artificial neural network with extracted features as an input. IV) Launching the trained artificial neural network to detect anomalous power consumption attack based on a threshold. In the simulations, the proposed method can detect anomalous power consumption attack with 74.25% accuracy in the worst case scenario. Also, four levels of wavelet transform make different features, so the proposed method has different performance.*

Keywords—Artificial neural network; wavelet transform; anomalous power consumption attack; smart grid; computer network security.

1. INTRODUCTION

The smart grid is becoming one of the largest-scale power delivery infrastructures in the few last decades. A smart grid uses advanced metering infrastructure (AMI) to provide bidirectional communication between smart meters and utilities providers [1]. The aims of the smart grid are real-time control, real-time measurement of power consumption, improved reliability, increased efficiency, reduced cost and improved security. However, the cyber layer in a smart grid opens up cyber threats and attacks that can ruin smart grid aims. On the other hand, transferred data between smart meters and utility centers are vital for smart grid infrastructure. Unfortunately, the transferred data can be manipulated by adversaries. Also, one of the most important attack in smart grid infrastructure is a data integrity attack that changes the smart meter reading (that is, a client side attack). In addition, this attack can change power consumption patterns in utility centers (that is, a server side attack).

In an anomalous power consumption attack, hackers try to manipulate data or signals by inserting, deleting, and changing control commands between consumers and controllers. The purpose of this attack is to deceive controller, control device or software to make a wrong decision, and lunch further attacks. One example of a data integrity attack is changing electricity bills and outages in the smart grid.

In addition, in 2010, a new malware, the Stuxnet worm, disrupted industrial nuclear systems using Programmable Logic Controllers (PLCs) [2]. This sophisticated worm has demonstrated that cyber-attacks effect not only computer networks and the Internet services, but also critical industrial processes.

Moreover, there is a key factor vulnerability in smart grids called a smart meter. A smart meter is an access point for bidirectional communications between its controller and a consumer. As a result, consumers can control their power consumption, and they can manage peak load by saving energy, reducing cost, and increasing reliability. There are two perspectives regarding smart grids: The power electric perspective and the communications perspective [1]. In the power electric perspective, the smart grid consist of generation, transmission, distribution and customers. In the communications perspective, the smart grid consist of wide area networks (WAN), neighborhood area networks (NAN), and home area networks (HAN). In Fig. 1 smart grid elements are shown [1].

ISBN: 1-60132-445-6, CSREA Press ©

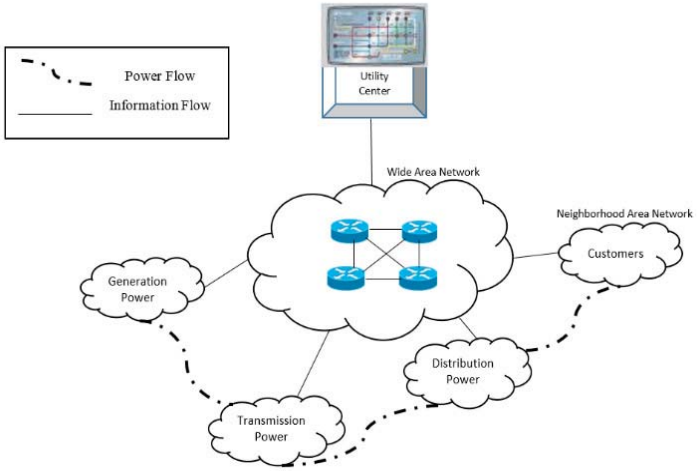


Fig. 1 Smart grid elements

In Fig. 2 [1], the utility center (in WAN level) shows the amount of utility for each subsystem NAN. The proposed method introduces an anomalous power consumption attack method for different levels of subsystems within the advanced metering infrastructure (AMI). When input usages of power in utility centers are not proportional with consumer's payments, a further technique is necessary to detect and locate the attack (in HAN level). The anomalous power consumption detection software was developed on routers at the HAN level of the smart grid infrastructure. The software on the router monitors the smart meters.

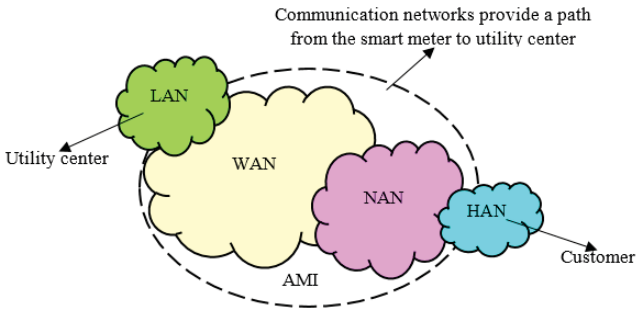


Fig. 2 Advanced metering infrastructure in smart grid.

In addition, normal energy usage patterns are different from hacker energy usage patterns. For example, a hacker injects data to a smart meter to evade the payment for power usage.

In this paper, an anomalous power consumption attack detection is introduced based on wavelet transform (WT) and artificial neural network (ANN). In the first step, the proposed method extracts features from power consumption of normal patterns and anomaly patterns by wavelet transform tool. Then, the ANN is trained to learn normal patterns and anomaly patterns from the extracted features. Therefore, ANN can distinguish between normal patterns and anomaly patterns of power consumption. Finally, the detection method can be launched for detecting anomalous power consumption attacks. If the result of cost function in ANN is greater than a threshold, the proposed technique detects an attack. Otherwise, if the result of cost function in ANN is less than a threshold, the pattern consumption is normal. The simulation results show that the proposed method can detect anomalous power consumption attacks with a high accuracy rate. In addition to proper accuracy rate of detection for a one hour attack, the proposed method has several additional advantages. First, it allows for fast detection anomalous behavior in real time. Second, it is sensitive to any attack duration. Third, it is continues learning which allows for improvement in detecting future attacks. Forth, it adapts to various environments.

The organization of this paper is as follows. Section 2 presents related works in smart grid. Section 3 presents wavelet transform. Section 4 presents ANN. Section 5 presents the proposed method. Section 6 and section 7 present the simulation and the result of simulation. Section 8 introduces concluding remarks.

2. Related Work

Liu et al. [3] presented one of dangerous attacks in smart grid, which is bad data injection (BDI). The aims of this attack are energy stealing on the consumer side, manipulating of the energy cost, modifying smart meter, taking control of power system, and breaking down power generation. This paper introduces an algorithm to find BDI attack which has four steps. In the first step, a large smart grid system is divided into several subsystems by a partitioning graph method. The aim of partitioning is enhancing the sensitivity of detection method. In the second step, each subsystems of smart grid power system are modeled with a mathematic equation. In the third steps, a BDI detection algorithm is lunched to detect the attack in each subsystem. Most of detection attack algorithm methods compute chi-square test and compare it with a threshold in each subsystem. If the chi-square test's result is less or equal than the threshold, then the detection attack algorithm which finds the subsystem is normal. On the other hand, if the chi-square test result is greater than the threshold, then the detection attack algorithm finds that the subsystem is suspected to bad data injection attack. In the fourth step, for locating the bad data, the subsystem's graph can be narrow to locate exactly suspicious region of bad data. Therefore, step two, three and four should be in a loop. Also, because of detecting and locating bad data injected, an iterative algorithm needs to find suspicious (desired) location. The paper needs high electricity power usage. This paper's proposed method works with low electricity power requirement.

Wrinch et al. [4] proposed anomaly detection by comparing the test building's results with ideal case

(threshold). This method finds anomaly behavior by extracting periodic energy and demand parameters of a power consumption of a building. In the first step, this method extracts information based on frequency domain (discrete Fourier transform) instead of time domain, and it finds energy from discrete Fourier transform. Then, it extracts periodic energy demand parameters of a building in frequency domain. Next, the proposed method extract periodic schedules. High energy periodical points present both periodic computer control and occupant activities. If energy in each harmonic is outside of a threshold, then the method can detect an anomaly (by simply observing a trend of increasing periodical demand energy ratios from the original baseline). The method can detect anomalies by energy demand in periodic building operations. This paper's proposed method can detect anomalies in various environments (not specific periodic electricity power usage).

Hu et al. [5] offers a taxonomy of attack based on cyber and physical schemes. According to this paper, power systems represent three phase current and voltage in a frame. All devices in smart grid are modeled in terms of differential equation that is device dynamics. Voltage, current and phase angle have to satisfied by algebraic constraint. According to this phase angle, this paper tries to detect attacks. For example, the proposed method can detect integrity attack when system is in unstable mode. The paper proposed a comprehensive scheme for physical and cyber-attack detection. Due to their architecture, there may be instances where the attackers attack the root node, and takes advantages of the bottleneck problem.

Mohammadi et al. [1] studied a framework for detecting attack in smart grid. The NAN-IDS consists of a distributed and hierarchical IDS which combines anomaly-based and signature-based methods. It extracts pattern and parameters of NAN network in smart grid such as data traffic, transmission power level, interval of transmission queries, request/reply pattern. The proposed NAN-IDS has three different kinds of IDS nodes: Field IDS, WAN IDS and central IDS. Field IDS monitors and collects trace data communication of the neighbor smart meters, and it reports of detected attacks to central IDS. WAN IDS is responsible for incoming and outgoing traffic from/to collectors. It reports the malicious nodes to the central IDS. Central IDS is responsible for making global decisions based on alarms and notifications coming from the WAN IDSs and field IDSs. The proposed IDS checks abnormality in their communication behavior. Finally, the proposed IDS makes final decision about anomaly behavior that is malicious attack, or it is just a transient failure. Then the IDS keeps the history of the monitored nodes to make accurate decisions in the future. The paper considers cyber domain issues in the smart grid, and physical domain issues are not considered. To provide a more comprehensive detection this present paper adds an analysis of physical domain parameters such as power consumption.

3. Wavelet Transform

Wavelet transform is a powerful framework and tool to analyze power consumptions more intelligently [7]. WT models power consumptions by time-frequency domain, and it uses long windows for high frequencies and short windows for low frequencies.

A short term power consumption prediction in the smart grid can be done by wavelet transformation. WT is proper to detect an irregular structure and an anomaly in signals [9]. WT can decompose a signal in coefficients, and it can localize an anomalous behavior in both time domain and frequency domain with different scales. WT is a multi-scale analysis. One of the most important properties of WT is its ability to adjust the size of a window to have a suitable resolution in time-domain and frequency-domain. Therefore, WT uses a narrow window function when it analyzes a high frequency; also, WT uses a wide window function when it analyzes a low frequency.

This property of WT makes it suitable to monitoring, supervising and detecting anomalous phenomena in dynamic signals, stochastic signals and dynamic systems [7]. In a time interval, when a signal changes rapidly, WT can zoom in the interval to extract more characteristics of a signal. Therefore, WT is sensitive to irregularities in signals, and WT is not sensitive to regular behaviors of signals. WT coefficients with anomalous events have larger magnitudes compared to WT coefficients without anomalous events. In addition, WT is a suitable tool for detecting anomalous because it reacts to the change to the first derivative (slope) of a signal, not to the change to the amplitude of a signal.

For detecting irregularities in a signal by computer, discrete wavelet transform (DWT) is used. One of the most important properties of DWT is its ability to analyze nonstationary signals. To detect an anomaly with low amplitude, short duration, fast rate of decay and rapid oscillation in a signal, the Daubechies wavelet is a proper choice as a mother wavelet for DWT. Additionally, fast calculation of WT coefficients is feasible by the Daubechies wavelet. Moreover, signal features and properties can be extracted by the Daubechies wavelet.

4. Artificial Neural Network

Artificial neural network is a learning model and efficient classifier, which has been influenced by brain functionality [6]. ANN is shown by an interconnected network of neurons that send messages to each other. This network of neurons has weights, and the weights can be adjusted based on previous experience. Therefore, this characteristic makes ANN able to learn from its environment.

Fig. 3 shows the structure of multilayer feed-forward ANN. The input layer obtains input from its environment and sends the input to the hidden layer. The purpose of the hidden layer is to connect the input layer and the output layer to extract more information and higher-order statistics from the

input layer. The response of a group of neurons of ANN is delivered by the output layer. Each circle node represents an artificial neuron and each line represents a connection from the output of one artificial neuron to the input of another artificial neuron.

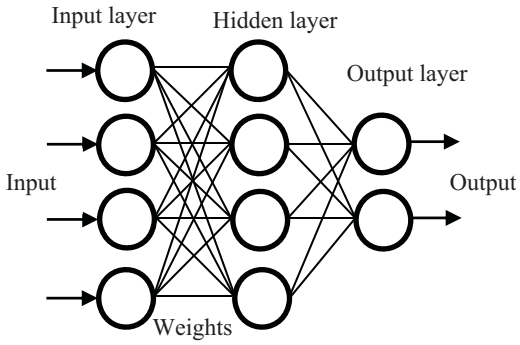


Fig. 3 Three-layered feed-forward ANN.

ANN has three important characteristics [7]. Firstly, a neuron is nonlinear, so interconnected networks of neurons are nonlinear. By this important property, modelling a nonlinear process is possible by ANN. For example power consumption is a nonlinear process that can be modeled by ANN. Secondly, ANN can map between inputs and outputs, so ANN is proper for pattern classification. Thirdly, ANN can be adapted to inputs, so weights in ANN are changed based on the environment. As a result, a neural network which operates in a specific environment can deal with a new environment easily.

On the other hand, ANN methods such as the backpropagation algorithm can be used to classify normal and anomalous behaviors in a signal (such as power consumption in the smart grid).

The input layer propagates input to the hidden layer, so the input values are multiple by a corresponding weight of each branch in the hidden layer and then summed [8]. The output of the hidden layer is propagated to the output layer. Error can be calculated as difference between output and desired output. The purpose of the backpropagation algorithm is minimize this error. Therefore, minimizing of the cost of error (cost function) can be defined as:

$$E(n) = \frac{1}{2q} \sum_{m=1}^{q} [d_m - y_m(n)]^2 \quad (1)$$

Where d_m defines the desired output of the m^{th} input training example, $y_m(n)$ represents the actual output of the neural network and q represents the total number of training examples.

5. Proposed Algorithm

The proposed scheme for detecting an anomalous power consumption attack is shown in Fig. 4, which can be described as below.

First of all, power consumption patterns of clients are obtained. Second, wavelet coefficients of power consumption are calculated by DWT. As a result, features of power consumption are extracted by the Daubechies wavelet transforms. To gain a higher anomaly detection rate, different Daubechies wavelet transform functions are used. Third, about 80% of the features are used to train the ANN system to extract normal and anomalous behaviors of power consumption patterns. Finally, the remaining 20% of the coefficients are used to test and classify normal power consumption and anomaly data patterns.

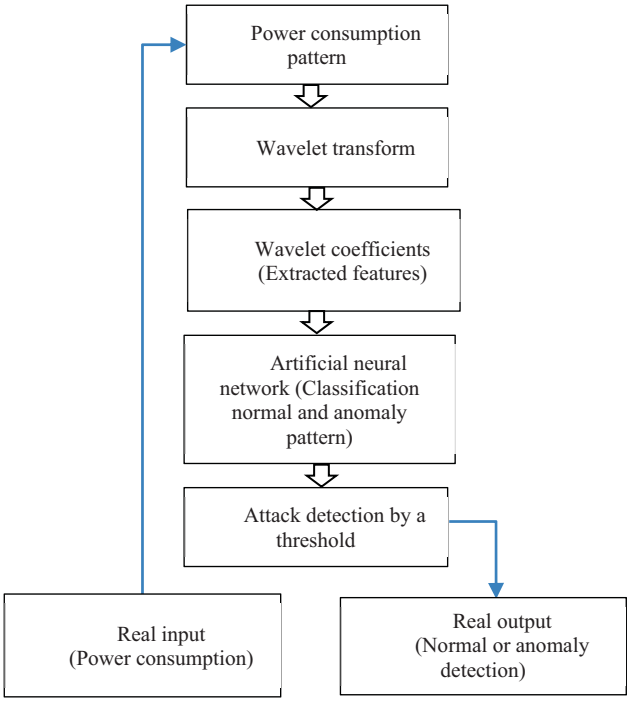


Fig. 4 Proposed scheme for detecting an anomalous power consumption attack.

6. Simulation

In this section, simulation of the proposed model for detecting the anomalous power consumption attack in the smart grid infrastructure is demonstrated. The code of the proposed method is written by software MATLAB R2014b. Also, MATLAB neural network toolbox such as graphical user interface "nntool" is used. Simulation of the proposed algorithm can be described as below.

In the first step, power consumption signals are generated synthetically (generated by computer). Therefore, one normal pattern of the power consumption of a customer is generated. The normal pattern has 64 samples per day. Then, elements of the normal pattern are permuted randomly between -0.3 and 0.4. Next, a set of normal patterns of power consumptions are created based on the first normal pattern of power consumption. Fig. 5 shows one sample of a normal

pattern of daily power consumption. Moreover, a set of anomaly patterns of power consumption are created based on the first normal pattern. Anomalous patterns are simulated by manipulating a smart meter with random duration of attack (for example, 50 minute duration of attack) and random starting point of attack (for example attack starts at 8:00 p.m.). Each set of normal and anomalous patterns of power consumption contain 1000 patterns. Next, a tag bit is used to separate the normal dataset from the anomaly dataset, so the normal dataset is tagged with zero, and the anomaly dataset is tagged with one. Finally, the normal dataset and the anomaly dataset are shuffled. As a result, the gained dataset contains 2000 patterns for evaluating the proposed method.

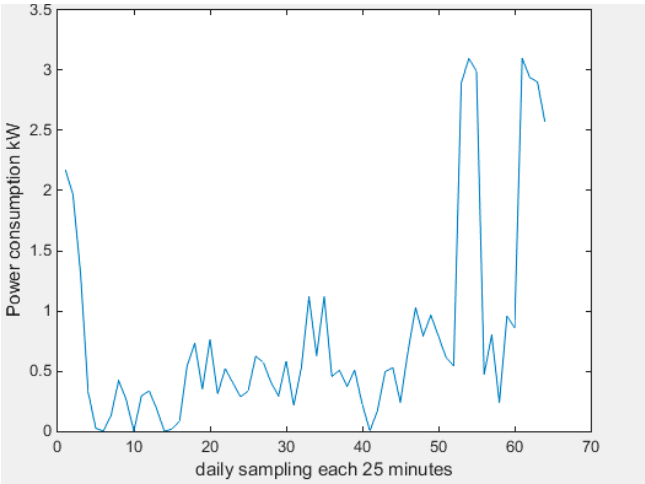


Fig. 5 A normal pattern of power consumption with 64 samples per day.

In the second step, features of power consumption patterns are extracted by the Daubechies wavelet transform 4 (db4 or D4). Different Daubechies D4 wavelet transform levels, ranging from level 3 to level 6, are applied to analyze power consumption. Different Daubechies D4 wavelet levels have different results and different sizes of output.

For example, when the input of the Daubechies D4 Wavelet Transform level 3 is 64 samples, the extracted output is 84 coefficients; when the input of the Daubechies D4 Wavelet Transform level 6 is 64 samples, the extracted output is 102 coefficients.

In the third step, about 80% of the coefficients are used to train the feed-forward backpropagation algorithm of ANN. The input size (the coefficients length) of ANN is varied based on the Daubechies wavelet levels. On the other hand, ANN has just one output, which is zero or one. Therefore, the existence of a threshold is necessary to classify the output in two groups of normal or anomaly. If the output is zero, the input data is normal. Therefore, there is not any attack, or the proposed method cannot detect the attack (false negative). On the other hand, if the output is one, the proposed method detects an anomalous behavior, or the proposed method wrongly suspects the input (false negative). The threshold for this algorithm is 0.5. Therefore, if the result of the cost function in ANN is less than 0.5, the output is zero, but if the result of the cost function in ANN is greater than 0.5, the output is one. Also, specific parameters are used to train the feed-forward backpropagation algorithm of ANN that is shown in table 1.

Table 1 : Specific parameters to train the feed-forward back propagation algorithm.

Parameters	Value
Number of layers	2
epochs	1000
max_grad	$1e^{-05}$
max_fail	60000

In the fourth step, the remaining 20% of the coefficients are used to test and classify normal power consumptions and anomalous data patterns. Also, in the testing step, the precision of the proposed method is evaluated, and the accuracy of detection is evaluated in terms of a true positive.

7. Results of Simulation

In this section, the results of applying the proposed model for detecting the anomalous power consumption attack in the smart grid infrastructure is demonstrated.

The true positive rate of the attack detection for the training and testing dataset is shown in Fig. 6 to Fig. 9. The duration of attack varies form 25 minutes to 750 minutes. Moreover, the worst case scenario for detecting the attack belongs to the attack duration of 25 minutes; Fig. 6 shows the lowest detection rate among four attack durations (Fig. 6 to Fig. 9).

The first column in Fig. 6 is extracted from the Daubechies D4 wavelet transform level 3, which is 71.88%, and it shows the true positive rate of the training dataset. The second column in Fig. 6 is 67.00%, which is extracted from the Daubechies D4 wavelet transform level 3, and it shows the true positive rate of the testing dataset. In addition, in Fig. 6, the highest detection accuracy belongs to the Daubechies D4 wavelet transform level 4 (77.75% for training dataset and 74.25% for testing dataset), and the lowest detection accuracy belongs to the Daubechies D4 wavelet transform level 3.

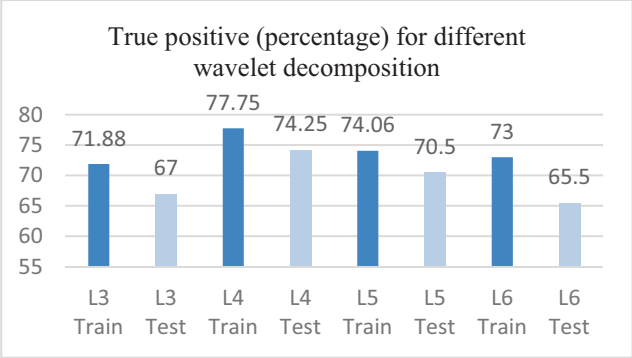


Fig. 6 True positive rate for a 25 minute (approximately) attack duration.

In Fig. 7 the highest detection accuracy belongs to the Daubechies D4 wavelet transform level 5, and the lowest detection accuracy belongs to the Daubechies D4 wavelet transform level 4.

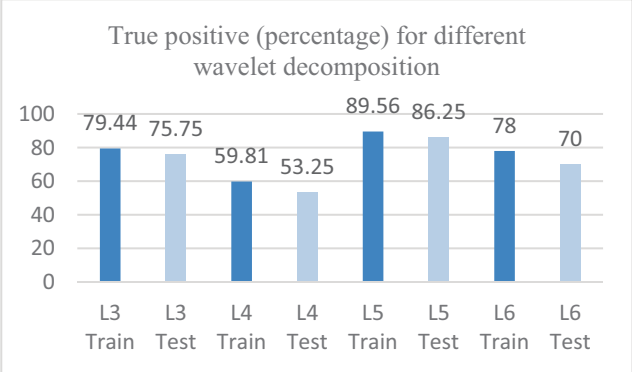


Fig. 7 True positive rate for a 50 minute (approximately) attack duration.

In Fig. 8 the highest detection accuracy belongs to the Daubechies D4 wavelet transform level 3, and the lowest detection accuracy belongs to the Daubechies D4 wavelet transform level 4.

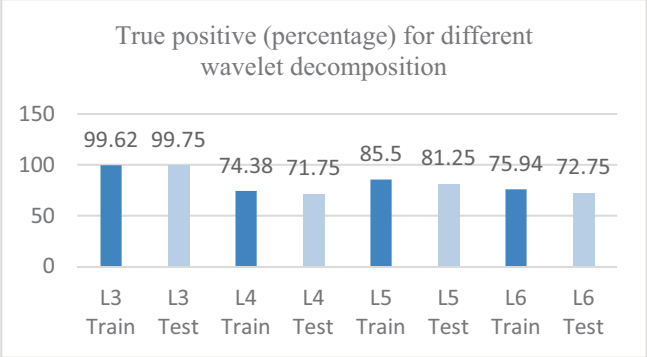


Fig. 8 True positive rate for a 150 minute (approximately) attack duration.

In Fig. 9 the highest detection accuracy belongs to the Daubechies D4 wavelet transform level 3, and the lowest detection accuracy belongs to the Daubechies D4 wavelet transform level 6.

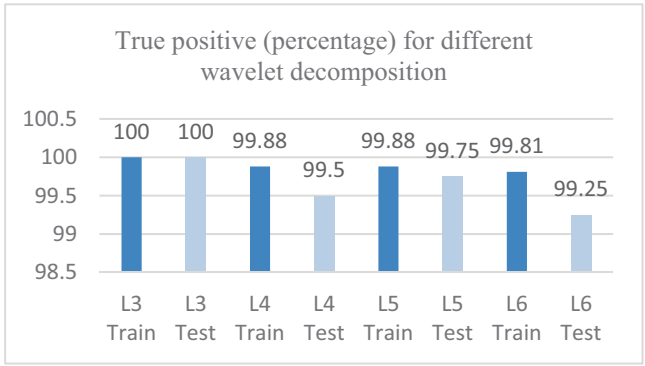


Fig. 9 True positive rate for a 750 minute (approximately) attack duration.

8. Conclusion

In this paper, a method is proposed to detect an anomalous power consumption attack. The fundamental idea of the method is to enhance the sensitivity of detection by using proper features and improving the training step in ANN of the proposed method to detect and locate the anomalous power consumption attack. The main procedure of proposed algorithm consists of the following steps: I) Finding normal and anomalous patterns of power consumption to train the proposed method. II) Applying WT to power consumption patterns to extract features. III) Training ANN with extracted features from step2. IV) Launching the trained ANN from step 3 to detect the anomalous power consumption attack based on a threshold.

In the simulations, the proposed method can detect the anomalous power consumption attack with 74.25% accuracy in the worst case scenario. Also, four levels of the Daubechies wavelet transform make different features, so the proposed method has different performance.

Future work will employ the algorithm that uses ANN to learn the most suitable features for this application.

References

[1] N. Beigi-Mohammadi, J. Misic, V. B. Misic, and Khazaei H. "A Framework for Intrusion Detection System In Advanced Metering Infrastructure".

Security and Communication Networks, pp. 1939-0122, 2012.
[2] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society , vol., no., pp.4490,4494, 7-10 Nov. 2011doi: 10.1109/IECON.2011.6120048
[3] T. Liu, Y. Gu, D. Wang, Y. Gui and, X. Guan, "A Novel Method to Detect Bad Data Injection Attack in Smart Grid," in Proc. IEEE INFOCOM Workshop on CCSES, 2013.
[4] M. Wrinch, T. H. M. EL-Fouly and S. Wong , "Anomaly detection of building systems using energy demand frequency domain anlaysis ," in proc. IEEE Power & Energy Society General Meeting, San-Diego, CA; 2012, pp. 1-6, Jul 2012.
[5] J. Hu, H. R. Pota, S. Guo, "Taxonomy of Attacks for Agent-Based Smart Grids," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1886-1895, Jul 2014.
[6] S. Haykin, Neural Networks - a comprehensive foundation, Prentice Hall, pp.1, 1994.
[7] F. Mo and W. Kinsner, "Wavelets and artificial neural networks in power system transient classification and short-term power load prediction," University of Manitoba, Winnipeg, MB, 2002.
[8] P. G. Kumar and D. Devaraj, "Network Intrusion Detection using Hybrid Neural Networks," 2007 Int. Conf. Signal Process. Commun. Netw., pp. 563-569, 2007.
[9] F. Mo and W. Kinsner, "Wavelet modelling of transients in power systems," Proc.IEEE Communications, Compurer & Power Conf., WESCANEX 097, IEEE, pp. 132-137, 1997.

Wormhole Attack Detection in Wireless Sensor Network Using Variance Fractal Dimension

Mohammad Nurul Afsar Shaon, Ken Ferens, Mike Ferens[1]
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada
[1]Gourdie-Fraser, Inc., Traverse City, Michigan, US
{shaonmna@myumanitoba.ca, ken.ferens@umanitoba.ca}

Abstract— Wormhole attack is one of the severe and critical security threats for wireless sensor networks (WSNs),as it can be started from legitimate sensor node and able to mount denial of service attack (DOS). According to the literature, neighborhood count is an important detection feature to identify wormhole attack in the network [1]. *The series of neighborhood counts is non-stationary and multi fractal time series, hence a fractal feature extraction method is required which is able to identify anomalies in the series of neighborhood counts. In this paper, variance fractal dimension is used as a technique to analysis the storied neighborhood count series in the base station. Variation of window length is also adopted to validate the calculation of variance fractal dimension of a data window. The Proposed detection scheme shows the promising performance in finding abnormal changes in the series that represent wormhole attack.*

Keywords—Variance fractal dimension; wormhole attack; wireless sensor networks.

I. INTRODUCTION

The wireless sensor network is simply a pool of self-directed devices organized into a mutually connected network. Sensors are usually autonomous and spatially distributed within a certain area to monitor targeted physical and environmental phenomena, such as temperature, sound and pressure. In WSNs, free frequency band and open architecture are used for supporting mission critical application in a hostile environment, thus they are highly prone to various security threats such as wormhole attack.

The wormhole attack is recognized as one of the most detrimental security threats for WSNs [2]. In WSNs, known communication channel is used so that the wormhole attack can be launched from the legitimate nodes without raising any security concerns. Wormhole attackers (node) are connected via virtual tunnel which can be set up in many ways (e.g. high quality hidden channel, packet encapsulation and high powered transmission) [3]. This direct low latency link is also known as *wormhole link* [4]. During this attack, a wormhole node E_1 captures packets around its radio communication range and sends them to another wormhole node E_2, via *wormhole link*. After receiving those packets, wormhole node E_2 broadcasts them within its communication area. By doing this, the attacker gains control over the data traffic passing through wormhole nodes by influencing other sensors to send data to the base station through wormhole nodes. As shown in Fig. 1, the E_1 and E_2 wormhole nodes, connected by *wormhole link*, capture the data packets from one location and replay them to another location.

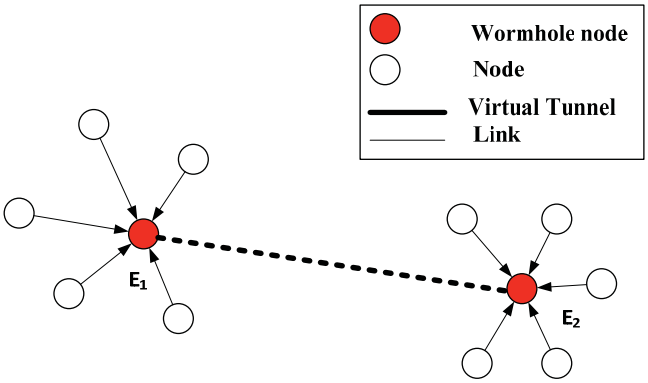


Fig. 1 Wormhole attack.

Moreover, this wormhole attack has severe impact on wireless sensor networks. It might destroy or hamper usual operation of the network by creating confusion and disruption in routing mechanism of the network [5]. The attacker may introduce DOS attack in the network by selective dropping of packets; manipulation of traffic; or modifying data packets without disclosing their identities.

Therefore, detection of wormhole nodes is an essential task for ensuring the security of a wireless sensor network. Most of the existing countermeasures use distance between

nodes, direction, and location abnormality among claimed neighbour nodes as detection features to fight against a wormhole attack. To gain a certain level of accuracy, many existing schemes have used complex and highly advanced devices, such as directional antenna [6], GPS [7], or ultra sound for distance measurement [8]. In fact, those special devices are very costly for a practical deployment. Some wormhole detection schemes based on hop count [9], node connectivity [4][10][11], or neighbourhood count [12][13] do not need any special hardware; however, they usually include a hardware supported scheme as a secondary approach. Furthermore, centralized statistical wormhole detection [12] may cause significant network and communication overhead in contrast to a distributed statistical approach [13]. In the network connectivity based wormhole attack detection schemes [4][10][11] , the positions of neighboring nodes are estimated from the received signal strength (RSSI) by each node and sends this information to the base station. By doing this, the network layout is determined by the base station and compared with the given network layout .This approach also causes significant amount of control packets flow to the base station. However, it is prone to the distance estimation errors especially for sparse network.

Furthermore, no. of neighbors and trust worthiness of a sensor node [14] are used as detection features in the neighbor based detection scheme. In the centralized method, each sensor node finds the number of neighbors within its communication region and sends this information to the base station. As the distribution of the sensor node is known, the base station computes the hypothetical distribution of the number of neighbors along with true distribution of the neighborhood counts. This process also creates significant amount of control data packet flow throughout the network and leads to the unexpected energy dissipation of sensor node. This process is also used as secondary approach with distance based scheme. In another neighborhood count based approach [1], Detector node takes count of neighbors at the each site it visited . In this approach, it may take significant amount of time to gather the dataset for applying machine learning technique such as ANN. During the initial stage of wormhole attack, the wormhole node doesn't drop the packets passing through it. The trust worthiness of a sensor as a detection feature may not detect the hidden wormhole attack.

Variance fractal dimension (VFD) is a quantitative measure of complexity of self-affine single or multi-featured time series. Unlike mono fractal time series, single integer value doesn't characterize the complexity of a multi-fractal time series. However, VFD is calculated over piecewise stationary sliding windows of the time series (over which VFD is calculated) continuously at different scale for the multi-fractal non-stationary time series. Moreover, this process produces the series of VFD of the sliding windows which is known as variance fractal dimension trajectory (VFDT). However, VFD scheme extracts the hidden features from overlapping or non-overlapping sliding windows of the time series that are used to solve pattern recognition and classification problems.

In this paper, we have proposed a VFDT based autonomous feature extraction scheme to detect wormhole attack in the network. This approach is able to trace abnormal changes in the non-stationary neighborhood count series without the help of any special hardware. Here, we have introduced a mobile node, called as detector node (D_N) that visits randomly chosen locations within the network area and collects neighborhood counts for each area visited. When the detector node D_N moves into a wormhole attack zone, the collected neighbor counts increase *abnormally* compared to a non-attack zone, in which the counts change *normally*. Those collected neighbor counts are sent to the base station and stored as the series of neighborhood counts. Since detector node sends collected neighborhood counts to the base station directly, this approach doesn't create significant amount of control data packets into the network. Once the data samples are gathered in the based station, VFDT based algorithm can be applied at that particular time. This technique provides an efficient way to detect the wormhole attack by measuring the complexity of the window of data samples. The flexible window size is adopted to maintain weak sense stationarity among the selected sliding windows. VFD measure would indicate the wormhole attack in the network as it reaches very close to the embedding (e.g. 2) dimension. Most significant advantage of the VFDT based algorithm is the computation of the VFD of a window can be done on the fly or batch mode. It would take less time to detect wormhole than other neighborhood based approach. Beside this, VFDT based algorithm detects wormhole attack with higher detection accuracy (almost 100 %) compare to other neighbor based algorithm.

The rest of this paper has been ordered as follows: Section II presents the literature survey on existing detection methods of wormhole attack. We talk about VFDT in section III. The proposed VFD based detection scheme is detailed in section IV. The evaluation results are discussed in Section V. Section VI includes concluding remarks and future scope of work.

II. RELATED WORK

Numerous detection schemes and counter measures have been proposed to confront the wormhole attacks in WSNs. In[7], the authors have developed a new method to detect wormhole attack using packet leashes. Two types of packet leashes are used: temporal packet leash and geographical packet leash. In the temporal leash (TL) method, a sender adds either packet sending time or expiration time, so that the receiver can verify if the packet has made a journey

unexpectedly too far based on the observed maximum transmission speed and time. In the geographical leash (GL), a sender includes its own location (using GPS) and sending time. Using GL, the maximum distance between the sender and receiver can be estimated by a receiver. This scheme can perform better if strict time synchronization and additional device like GPS are provided, which are not always possible or practical to include in low resource sensor nodes.

In [6], a new idea has been introduced to detect a wormhole attack. A directional antenna is attached to each sensor node to detect a wormhole node. According to the authors, if a sensor sends a packet in a given direction, its receiver will receive it in the opposite direction. Therefore, the authenticity of a neighbor can be verified by their sending and receiving directions. This scheme appears to require additional hardware (i.e., a directional antenna), which may not be possible or practical.

The method in [4] aims to detect a wormhole node by looking at the connectivity graph for forbidden substructures. Two non-neighbor nodes might have at most f_k common independent *k*-hop neighbors; an attack is spotted if the opposite happens. Compared to a dense network, forbidden substructures are very hard to find in a sparse network.

Another category of wormhole detectors has been proposed based on the investigation of the statistical parameters of network, such as the number of neighbors and hop count. In [12], the statistics of total hop count and neighbor information are monitored by the base station. If the total number of hop counts decreases dramatically or whether the neighborhood count of all nodes increases over a threshold, presence of a wormhole node is declared. However, this scheme causes significant communication and co-ordination overhead, due to its centralized design.

In [13], another statistical approach is proposed, known as SWAN, in which each sensor node captures a recent number of neighbor**s**. A wormhole attack is identified if the current number of neighbors exhibits an unusual increase, compared to the previous neighborhood counts taken outside of the wormhole zones. This is a distributed approach so that it doesn't cause any overhead, unlike a centralized approach. However, both schemes [12] and [13] have been designed for and perform better in a uniformly distributed network, but their performance is in question for networks in which sensors are distributed non-uniformly.

III. VARIANCE FRACTAL DIMENSION

Fractal analysis is the measure of complexity of a single or multi featured self-affine time series [15] . If we are interested to analyze the complexity of the single or multi-featured time series more efficiently in real time, variance fractal dimension is the best analysis technique to do it. Variance fractal dimension is not an equivalent matter like measuring the variance of the process. Thus, Variance fractal dimension is a special class of fractal analysis to measure the complexity where second moment of statistics (variance) of the data samples is used at multiple scales.

Let us assume that $s(t)$ represents a time series which is either continuous or discrete. The variance of this series σ^2 of its amplitude increments over time increment is proportional to the time increment according to the specific power law [15]. The power law is given below.

$$Var[S(t_2) - S(t_1)] \sim |t_2 - t_1|^{2H} \quad (1)$$

Where H is called as the Hurst exponent and is constrained in between 0 and 1. If $H = 0$, the process shows negative correlation. i.e. The process exhibits no persistency in its trend (i.e. White noise). If $H = 1$, the process exhibits long term persistency in its trend (as black noise). If $H = 0.5$, short term persistency is observed in the process. Thus, this process is categorized as fractal Brownian motion [16]. If we set

$$\Delta t := |t_2 - t_1| \quad (2)$$

$$(\Delta S)_{\Delta t} := S(t_2) - S(t_1) \quad (3)$$

Then Hurst exponent, H is calculated using following equation

$$H = \frac{1}{2} \lim_{\Delta t \to 0} \frac{\log[var((\Delta S)_{\Delta t})]}{\log(\Delta t)} \quad (4)$$

As variance fractal dimension is related to the H and embedding dimension, E so that their relationship can be expressed as

$$D_\sigma = E + 1 - H \quad (5)$$

Where D_σ is the variance fractal dimension bounded between 1 to E [17]. For a single featured time series, above mentioned equation can be rewritten as

$$D_\sigma = 2 - H \quad (6)$$

Variance Fractal Dimension Trajectory

Variance fractal dimension can be calculated continuously for non-stationary self-affine time series. In order to do it, time series is required to be divided into several windows such that (weak sense) stationarity property of each window is preserved [18]. In addition to, Windows can be chosen as overlapping or non-overlapping fashion. Then, VFD is calculated for every sliding window at multiple scale

continuously. Then the plot of the VFD of each window is stated as variance fractal dimension trajectory (VFDT). Moreover, Global variance fractal dimension (GVFD) can be calculated from the VFDT by taking the arithmetic mean of the measured VFD values [15]. Some points are required to keep in mind while calculating VFDT.

1) Weak sense stationarity must be preserved for each window.

2) Variance fractal dimension should be bounded between 1 and embedding dimension, E. If it exceeds the limits of E and 1, the calculation procedures and window size selection procedures are required to be revisited.

3) Saturation points and outliers in the log-log plot are needed to be eliminated before calculating slope.

IV. PROPOSED ALGORITHM

The proposed algorithm is applied on the series of neighbor counts collected by the mobile sensor node.

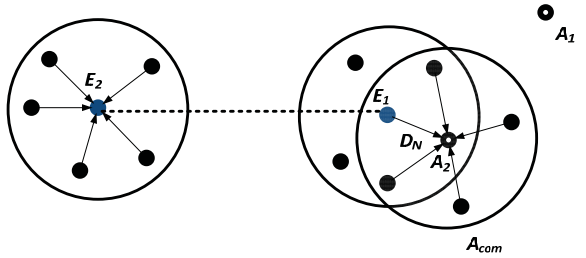


Fig. 2 Impact of wormhole attack.

A mobile sensor node, known as detector node (D_N), is deployed in an area where sensor nodes are distributed uniformly. The detector node (D_N) moves around this deployed area and collects a neighborhood count at each site it visits. When it reaches into the communication range of a wormhole node, the counted number of neighbors would increase abnormally sharply. The counts are gathered and recorded in the base station for further analysis. For instance, Fig. 2 shows how the neighbor counts are affected by wormhole nodes. The detector node (D_N) moves from one location A_1 to another location A_2. Then the detector node (D_N) will receive the new neighbor beacon message from each sensor node within its communication range, A_{com}. At the same time, sensors, around the E_2, also send beacons via E_1 as they are connected through a virtual tunnel

A. Algorithm:

1) Detector node (D_N) collects the neighborhood count at each site it visits and sends it to the base station.

2) Base station receives and stores the samples captured by detector node (D_N) as the series of neighborhood counts.

3) Select the appropriate window size to the calculate VFD in batch mode. Vary the window length to maintain stationarity properties among the sliding windows if it is necessary [16].

$$window\_size = nL \qquad where\ n = 1,2, \ldots\ldots$$

Where L is the minimum window size for calculating VFD.

4) Select the first window of data samples. Let's assume that there are N data samples in a window.

5) Select the largest vel size, K_H in such a way at least 30 vels are required to cover the N samples.

$$K_H \geq \left\lceil \frac{\log N}{\log b} \right\rceil$$

Where b is the number base. In our research, we have used $b=2$.

6) Select the lowest cover size such that it contains at least 2 samples.

$$K_L \geq 1$$

In our case, we have set $K_L = 1$

7) Run the main loop from K_H to K_L

a) Each stage of K, calculate the number of samples per vel and number of vels required to cover the N samples.

$$n_k = b^k$$
$$N_k = \left\lfloor \frac{N}{n_K} \right\rfloor$$

Where n_k and N_k represent the number of samples in each vel and number of vels covers N samples in K^{th} stage.

b) For each K^{th} stage, calculate variance by following expressions

$$Var(\Delta S)_K = \frac{1}{N_K - 1}[\sum_{i=1}^{N_K}((\Delta S)_{iK})^2 - \frac{1}{N_k}\left(\sum_{i=1}^{N_K}(\Delta S)_{iK}\right)^2]$$

c) Calculate coordinates for log-log plot.

$$X_K = \log(n_K)$$
$$Y_K = \log(Var(\Delta S)_K)$$

d) Continue this procedure until K reaches to K_H.

e) From the set of coordinates, calculate the slope from log-log plot and Hurst exponent to determine VFD of the selected sliding window.

8) Select another sliding window (non-overlapping) and repeat steps 5 to 7.

9) If the VFD of a window is greater than 1.90 approximately or very close to embedding dimension, then existence of wormhole attack is declared.

V. SIMULATION AND RESULTS

In this section, the proposed VFD based algorithm has been applied to extract the hidden features from the acquired series of neighborhood counts (calculated at base station) to detect wormhole attack in the network. In the first part of the simulation, we have evaluated the VFDT for both series that captured with the presence of wormhole node and without the presence of wormhole node. After that, we have investigated the impact of wormhole attack on global variance fractal dimension.

In this simulation, 300 sensor nodes are spatially distributed within 500 meters by 500 meters area. Radio range of each sensor node is 50 meters. A mobile sensor node is deployed to collect neighborhood count within this given area whose radio range is same as other sensor nodes. A pair of wormholes is placed on a location of 150 meters by 150 meters and 300 meters by 300 meters The random waypoint model is used as the mobility model for the simulation [13].

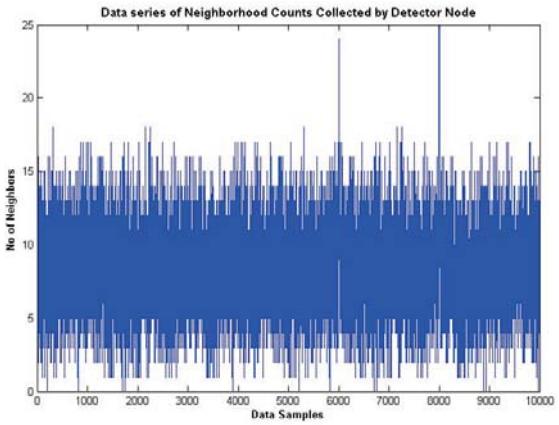


Fig. 3 Series of neighborhood counts (with presence of wormhole attack)

Fig. 3 represents the series of neighbor counts collected by mobile sensor node. However, we have observed peak values of neighborhood count from sample no. 6001 to 6015 and sample no. 8000 to 8017 in this time series when detector node (D_N) visits wormhole affected zone.

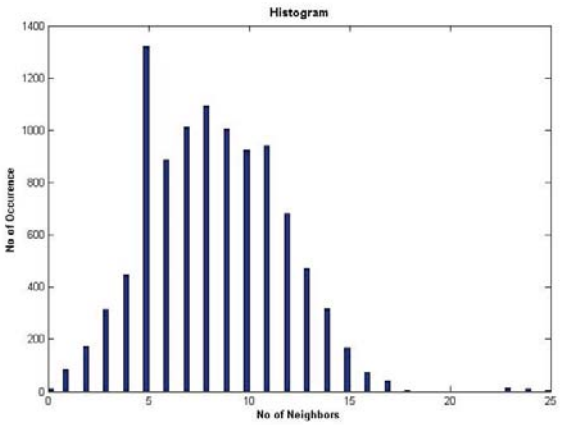


Fig. 4 Histogram of series of neighborhood ounts

As shown in Fig. 4, the histogram has been obtained by selecting the bin size of 100 to observe the number of occurrences of neighborhood counts from the time series.

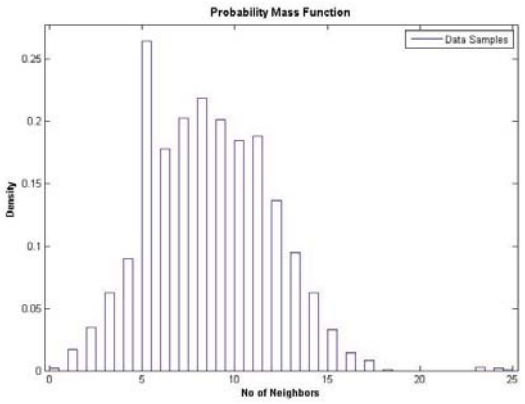


Fig. 5 Probability mass function of the Series of neighbor counts

Fig. 5 represents probability mass function (Pmf) of this data series. Pmf has been obtained by normalizing the histogram. We have observed rich variation in the probability distribution. This distribution is similar as bell shaped curve and stochastic in nature.

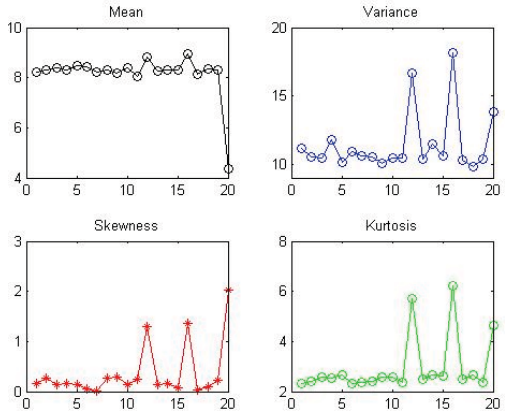


Fig. 6 Stationarity condition for window size 512.

According to the literature, the minimum window length for applying VFDT is 512. However, we have to make sure that the data set must be weak sense stationary for the selected window size. According to the Fig. 6, the data set is not stationary for the window length 512.Therefore, the flexible window length is adopted.

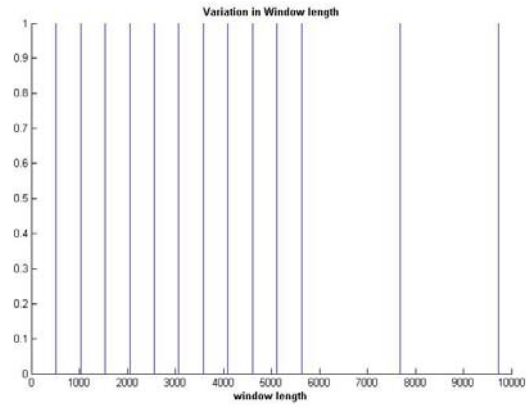


Fig. 7 Variation in window length

As shown in the Fig. 7 , the variation in window length has been applied to ensure the stationarity of the second moment of statistics among non-overlapping windows. Here, in the 12th window, initial sample size was 512 which starts at sample number 5633. Another 1536 samples were added to ensure the stationarity of the second order statistics (after the increment, total window size is 2048). Same thing happens for the window starts at sample no.7681 (13th window).

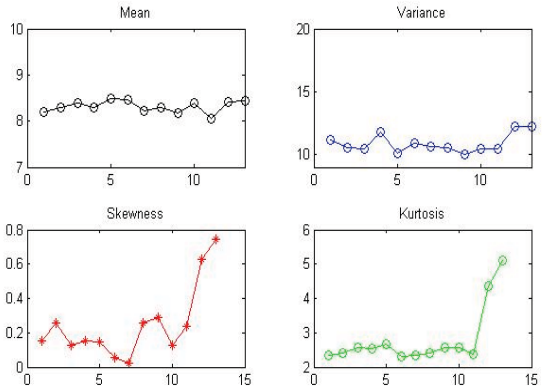


Fig. 8 Stationarity check after adospting flexible window length.

According to the Fig. 8, since the flexible window size is adopted, the data set becomes weak-sense stationary. When attack data samples appear in the series, it makes the data set non-stationary. In others words, the size of a particular window has been adjusted when the attack data samples appears in that particular window. That's why, it can be said that the data set is non-stationary due to the presence of attack samples and this VFDT based approach is able to detect wormhole attack from the non-stationary data series.

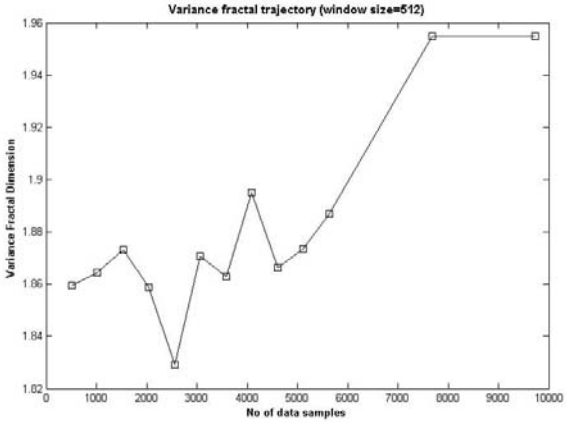


Fig. 9 VFDT of the series of Neighbor couts (with presence of wormhole attack)

Fig. 9 represents the variation of VFD of all non-overlapping sliding windows. As we know, VFDT remains constant for mono-fractal time series. In this case, VFD is not constant for all other sliding windows. Therefore, it is a clear indication that this time series is a multifractal time series. According to the Fig. 3 and Fig. 9, VFDT has followed the trend of time series. Moreover, it has been observed that, when time series represents 'no attack' data samples, the VFD fluctuates approximately from 1.82 to 1.90. Furthermore, when wormhole attack infected samples appear in the time series, (at 12th and 13th window), VFD starts increasing and reaches very close to embedding dimension of 2 (i.e. around 1.96). Thus, wormhole attack could be identified if the VFD of any sliding window surpasses the value 1.90.

Furthermore, the impact of the wormhole attack on GVFD has been analyzed. To do so, we considered two cases: with the presence of wormhole nodes and without the presence of wormhole nodes.

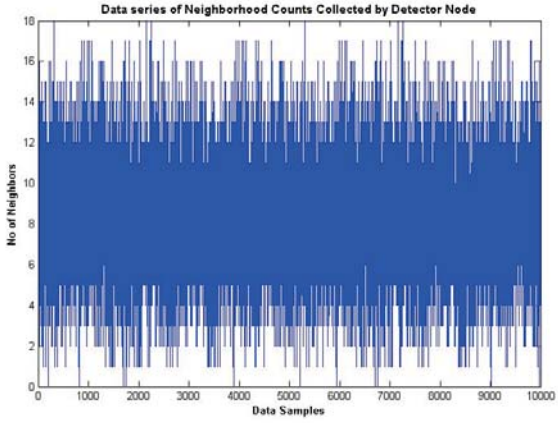


Fig. 10 Series of neighborhood Counts (without presence of wormhole attack)

Fig. 10 shows the series of neighborhood count which contains no malicious data samples. Sensor node distribution and it's radio range are same as stated before.

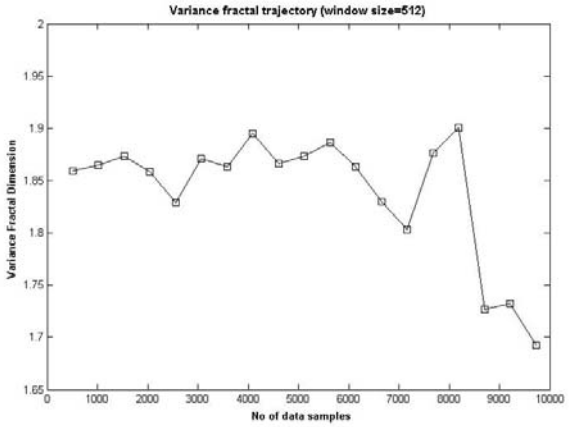


Fig. 11 VFDT of the series of Neighbor couts (without presence of wormhole attack)

Fig. 11 exhibits VFDT of the series which doesn't contain any count of neighbor that represents wormhole attack. Non overlapping window of 512 samples is used to evaluate VFDT of the series shown in Fig. 10 . The trajectory shows the values of VFD fluctuate within 1.7 to 1.90.

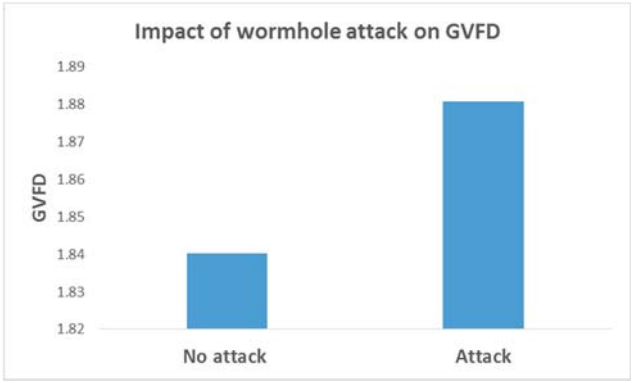


Fig. 12 Impact of wormhole on GVFD

According to the Fig. 12, GVFD is measured 1.8403 when time series represents 'no attack' data samples shown in Fig. 10. GVFD is increased and reached at 1.8808 when time series contains some neighbor counts that indicate wormhole attack shown in Fig. 3. With the presence of the wormhole attack, 2.201% increment is observed in GVFD. This increment of GVFD also indicates the presence of the wormhole attack in the network.

VI. CONCLUSION

This paper represents the VFD based feature extraction scheme to detect wormhole attack in the network; for the time series that contains malicious data samples which is non-stationary and stochastic in nature. However, variation of window length has been adopted to ensure the stationarity of the second order statistics among all non-overlapping windows. If The VFD reaches very close to embedding dimension of the time series (i.e. 2), wormhole attack can be identified. In future, the performance of VFDT based detection scheme will be evaluated for non-uniform sensor distribution.

VII. ACKNOWLEDGEMENT

I would like to thank Dr. Witold Kinsner, Department of Electrical and Computer Engineering, University of Manitoba, for their guidance and humble support during this study.

REFERENCES

[1] Mohammad Nurul Afsar Shaon and Ken Ferens, "Wireless Sensor Network Wormhole Detection using an Artificial Neural Network," *ICWN*, pp. 115–120, 2015.

[2] Y. Hu, A. Perrig, and D. B. Johnson, "Wormhole Attacks in Wireless Networks," vol. 24, no. 2, pp. 370–380, 2006.

[3] M. E.-S. Marianne Azer, Sherif El-Kassas, "A Full Image of the Wormhole Attacks Towards Introducing Complex Wormhole Attacks in wireless Ad Hoc Networks," *Int. J. Comput. Sci. Inf. Secur.*, vol. 1, no. 1, pp. 41–52, 2009.

[4] Y. Xu, G. Chen, J. Ford, and F. Makedon, "Detecting wormhole attacks in wireless sensor networks using connectivity information," *Crit. Infrastruct. Prot.*, vol. 2006, 2007.

[5] R. Song, P. C. Mason, and M. Li, "Enhancement of frequency-based wormhole attack detection," *Proc. - IEEE Mil. Commun. Conf. MILCOM*, pp. 1139–1145, 2011.

[6] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," *Netw. Distrib. Syst. Symp. NDSS*, no. February, pp. 1–11, 2004.

[7] Y.-C. Hu, a. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," *IEEE INFOCOM 2003. Twenty-second Annu. Jt. Conf. IEEE Comput. Commun. Soc. (IEEE Cat. No.03CH37428)*, vol. 3, no. C, pp. 1976–1986, 2003.

[8] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," *Proc. 2003 ACM Work. Wirel. Secur. WiSe 03*, vol. 0, no. Section 2, pp. 1–10, 2003.

[9] N. Song, L. Qian, S. Ning, Q. Lijun, and L. Xiangfang, "Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach," *Parallel Distrib. Process. Symp. 2005. Proceedings. 19th IEEE Int.*, p. 8 pp., 2005.

[10] M. a Azer, S. Member, S. M. El-kassas, M. S. El-soudani, and S. Member, "An Innovative Approach for the Wormhole Attack Detection and Prevention In Wireless Ad Hoc Networks," pp. 366–371, 2010.

[11] Y. Zhou, L. Lamont, and L. Li, "Wormhole attack detection based on distance verification and the use of hypothesis testing for wireless ad hoc networks," *2009 IEEE Mil. Commun. Conf. MILCOM 2009*, 2009.

[12] L. Buttyán, L. Dóra, and I. Vajda, "Statistical Wormhole Detection in Sensor Networks," *Secur. Priv. Adhoc Sens. Networks*, pp. 128–141, 2005.

[13] S. Song and H. Wu, "Statistical Wormhole Detection for Mobile Sensor Networks," pp. 322–327, 2012.

[14] and Ý. G. Özdemir, S., M. Meghdadi, "A time and trust based wormhole detection algorithm for wireless sensor networks," *3rd Inf. Secur. Cryptol. Conf.*, vol. 94, no. 20, pp. 1–5, 2008.

[15] W.Kinsner, "Fractal and Chaos Enginnering ,Class Notes," p. 900, 2010.

[16] M. S. Khan, K. Ferens, and W. Kinsner, "A Polyscale Autonomous Sliding Window for Cognitive Machine Classification of Malicious Internet Traffic," *World Comp*, pp. 1–7, 2015.

[17] W. Kinsner and W. Grieder, "Amplification of signal features using variance fractal dimension trajectory," *Proc. 2009 8th IEEE Int. Conf. Cogn. Informatics, ICCI 2009*, pp. 201–209, 2009.

[18] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Applications of Variance Fractal Dimension: a Survey," *Fractals*, vol. 22, no. 01n02, p. 1450003, 2014.

Network Intrusion Detection Using Machine Learning

Md Nasimuzzaman Chowdhury and Ken Ferens, Mike Ferens[1]
Department of Electrical and Computer Engineering University of Manitoba
Winnipeg, Manitoba, Canada
[1]Gourdie-Fraser, Inc., Traverse City, Michigan, US
{chowdhmn@myumanitoba.ca, ken.ferens@umanitoba.ca}

Abstract— *In the network communications, network intrusion is the most important concern nowadays. The increasing occurrence of network attacks is a devastating problem for network services. Various research works are already conducted to find an effective and efficient solution to prevent intrusion in the network in order to ensure network security and privacy. Machine learning is an effective analysis tool to detect any anomalous events occurred in the network traffic flow. In this paper, a combination of two machine learning algorithms is proposed to classify any anomalous behavior in the network traffic. The overall efficiency of the proposed method is dignified by evaluating the detection accuracy, false positive rate, false negative rate and time taken to detect the intrusion. The proposed method demonstrates the effectiveness of the algorithm in detecting the intrusion with higher detection accuracy of 98.76% and lower false positive rate of 0.09% and false negative rate of 1.15%, whereas the normal SVM based scheme achieved a detection accuracy of 88.03% and false positive rate of 4.2% and false negative rate of 7.77%.*

Keywords—Intrusion Detection; Machine Learning; Support Vector Machine, Supervised Learning

1. INTRODUCTION

Network Security maintenance is one of the major safety concerns for neutralizing any unwanted activities. It is not only for protecting data and network privacy issues but also for avoiding any hazardous situations. From January through June 2010 Microsoft security intelligence report shows that the infection trends are still increasing on average around the world at a higher rate [1]. For decades, Network security is one of the major issues and different types of developed systems are being implemented. Network intrusion is an unauthorized activity over the network that steals any important and classified data. Also sometimes it's the reason of unavailability of network services. The unexpected anomaly occurs frequently and a great loss to internet cyber world in terms of data security, the safety of potential information's etc. Therefore, the security system has to be robust, dependable and well configured. Principally it is of two types on network intrusion detection. One is signature based and another is anomaly based detection system. Signature based detection system involves analyzing network traffic for a series of bytes or packet sequences known to be an anomaly. A major disadvantage of this detection scheme is that signatures are comparatively fair easier to develop and understand if one knows what network behavior need to be identified. Signature based type detection also has some disadvantages. A signature needs to be created for each attack and they are able to detect only those attacks. They are unable to detect any other novel attacks as their signatures are unknown to the detection scheme.

The Anomaly based type detection scheme concept is based on analyzing the characteristic of the network behavior. This type detection has the capability to detect anomaly behavior by analyzing the high volume traffic, a surge in traffic from a specific host or to a specific host, load imbalance in the network [2]. One disadvantage of this kind of scheme is that if the malicious behavior falls within normal network behavior then it's not detected as an anomaly. Major Advantages over the signature based is, a new attack for which a signature does not exist can be detected if it behaves differently from normal traffic behavior patterns. For data confidentiality, classified data security and for preventing unauthorized access detection of intrusion is an essential task for ensuring the security of the networks.

There are several types of method proposed for network intrusion detection. The anomaly network intrusion detection is a major part of network security [3], [4]. Sometimes the behavior of the anomaly seems to be similar as normal data usage [5]. One problem in anomaly detection refers to the issue of classification problem that how to make a distinction between normal

and abnormal activities in an effective and efficient way.

Presently machine learning system has been extended for implementing effective intrusion detection system. Machine learning methods are very functional and improved in current intrusion detection. In particular, support vector machines [6], neural networks [7], decision trees seems to have efficient significant schemes in anomaly detection systems to improve the classification performance and speed.

In this paper, an new algorithm is proposed using a combination of two machine learning methods Simulated Annealing & Support Vector Machine that can detect any anomalous behavior of the network and can able to classify between normal and abnormal behavior. It doesn't require any hardware specifications and can be used for pattern recognition of the malicious behavior.

2. Support Vector Machine

Support vector machines (SVM) [8], a type of machine learning method; capable of being performing a range of classification tasks. It's also a set of related supervised learning methods that can analyze data and recognize patterns.

SVMs have been evolved to give a standard generalized performance to solve wide range classification and pattern recognition problems such as handwritten character recognition [9], face detection [10], pedestrian detection [11], and text categorization.

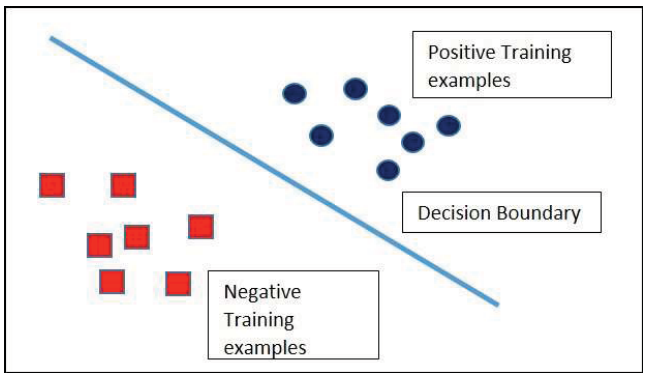


Figure 1: A linear Support Vector Machine.

Considering a training dataset $\{X_i, Y_i\}_{i=1}^{n}$, where X_i represents the input vector of svm that contains the n dimensional input features and $Y_i \in \{+1, -1\}$ represents the output. $Y_i = 1$ Denotes the positive group of training samples and $Y_i = -1$ denotes the negative training samples.

The decision surface in the form of hyperplane is defined as

$$W.X + b = 0 \quad (1)$$

Where,

W= Weight Vector
b = The bias

Linear SVM maximizes the geometric margin of training dataset.

$$max_{w,b}\ C$$

s.t

$$Y_i \frac{1}{||w||}(W.X_i + b) \geq C, i = 1. \ldots n \quad (2)$$

Where C is called the regularization parameter.

Any solution that can be found within the constrain boundary, any positively scaled multiples will satisfy them too. So if

$||w|| = 1/C$, the linear SVM can be formulated as

$$\min||x|| \leftrightarrow \frac{\min\ 1}{2\ ||x||^2}$$

s.t

$$Y_i(W.X_i + b) \geq 1, i = 1, \ldots \ldots . n \quad (3)$$

With this settings a margin around the linear decision boundary can be shown at a higher dimension

3. Selected Dataset and Feature Description

The dataset used for this research paper were chosen from the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). In this dataset, a hybrid of real modern normal activities and attack behaviors were generated. This dataset contains total forty-seven features and also contains over 2 million sample data [12] [13].

Feature selection is the most important step for network intrusion detection. Features play an important role to achieve classification accuracy which improves the effectiveness and also the efficiency of an intrusion detection system. In this paperwork, to detect an anomaly and to achieve the highest detection accuracy with the shortest possible time, a set of three features were randomly selected each time by the algorithm. Then SVM was performed among those randomly selected features to find the detection accuracy, false positive, false negative and time taken for anomaly detection.

The following table describes a few randomly selected features combinations that have been used for detecting anomalous and normal behavior in the network data traffic [12] [13].

Table 1: Features

Combination Number	Features in this Combination
1	**a**. Source transmission control base sequence number **b**. Source TCP window advertisement value **c**. Source and Destination IP address equal and port numbers
2	**a.** If the ftp session is time series is accessed by user and password **b**. Number of flows that has a command in ftp session. **c**. Number of connections in same source and destination address in past 100 connections
3	**a.** Source transmission control connection setup round-trip time **b**. Source and Destination IP address equal and port numbers **c**. No. of connections in same destination address and source port in past 100 connections
4	**a.** Destination TCP base sequence number **b**. Source and Destination IP address equal and port numbers **c**. No. of connections in same destination address and source port in past 100 connections
5	**a.** From the Source to destination time to live value while the packets are alive **b**. Source TCP window advertisement value **c.** No. of connections that contain same service and source address in previous 100 connections
6	**a.** If the ftp session is time series is accessed by user and password **b**. No. of connections in same source address and destination port **c**. No. of connections in same source and destination address in past 100 connections
7	**a.** Record start time **b**. Source inter packet arrival time (mSec) **c**. No. of connections that contain same service and source address
8	**a** From the Source to destination time to live value while the packets are alive **b**. Destination to source packet count **c**. No. of connections in same destination address and source port in past 100 connections
9	**a.** From the Source to destination time to live value while the packets are alive **b.** Destination inter packet arrival time (mSec) **c**. No. of connections in same source address
10	**a.** Destination packets retransmitted or dropped **b**. Destination to source packet count **c**. Source jitter (mSec)

4. PROPOSED MACHINE LEARNING ALGORITHM

The proposed algorithm is based firstly on simulated annealing that makes random combinations of 3 features at a time and then SVM is applied on that feature combination that is able to detect anomalous behavior from the internet data traffic. The details of the proposed algorithm is given below:

1. Define the number of features, K from the dataset.

2. Select n features among K features using random combination where $n \in K$
3. Run SVM on n featured training examples
   a) Select the total number of N data samples (n featured) to run the SVM.
   b) Select SVM parameter (Gamma, coef θ, nu etc.)
   c) Select $n \times N$ data sample for training and save the data on T_{train} dataset
   d) Select $n \times M$ data samples for testing and save it in T_{test} dataset
   e) Using T_{train} train the SVM
   f) After training, the learning performance of SVM is evaluated. Using T_{test} , detection accuracy, false positive rate, false negative rate and time taken to run the model are measured.
4. Repeat the procedure from 2-3 until highest detection accuracy, low false positive and false negative rate are achieved. After this the randomly selected n features are stored.

At first, proposed scheme defines the number of features in the dataset. Furthermore, n features are selected using simulated annealing to generate a combination of three features among the total 47 features to see which combination of the features is relevant to achieve highest detection accuracy. After that, the algorithm selects the N number of data samples which contains both normal and abnormal data traffic pattern to run the SVM-based scheme. Then it randomly selects n number of appropriate features for detecting abnormal behavior from network data traffic. Before running the algorithm parameters of the proposed algorithm such as $gamma, coef\ \theta, nu\ etc$ were initialised. After mixing up the dataset, $n \times N$ data samples are selected so that the SVM can learn the dataset without any bias. These samples are stored in T_{train}that will be used for training purpose. Similarly $n \times M$ data samples are chosen and stored in $T_{test}$to verify the learning performance of the SVM-based detection scheme. After performing the learning procedure of SVM, detection accuracy, false positive rate and time to run the algorithm are measured. The whole process is repeated from steps 2-3 until all possible combination (kCn) of features are evaluated to analyze which combination of features provide the highest detection accuracy and lower false positive and false negative rate.

To be noted that the simulated annealing is programmed in a way that it will not generate a similar or repeated combination of features. For example, among the 47 features if a combination of feature number [1, 13, 27] is generated randomly then it will not generate a combination like [27, 13, 1] or [13, 1, 27].

5. Simulation & Results

In this section, the simulation is performed to validate the performance of the proposed algorithm. In the first phase, the experiment is conducted and analyzed that whether the proposed machine learning algorithm is able to differentiate between normal and anomaly behavior or not. The percentage of detection accuracy, false positive, false negative and time have been evaluated for the proposed method. Finally, we have investigated the performance of the proposed algorithm by randomly selecting 3 features at a time among 47 features and apply SVM on a different number of feature combinations. The whole experiment has been conducted using lib-linear machine learning tool.

From the dataset, 150,000 samples are selected randomly which contains 75,000 normal and 75,000 anomaly samples. 80% of the number of samples are used for training and rest of them are used for testing the algorithm.

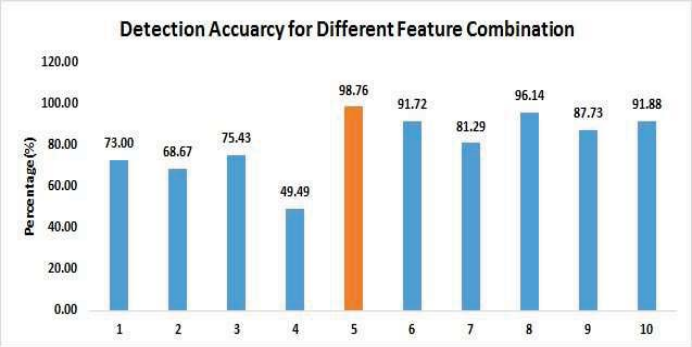


Figure 2: Percentage of Detection Accuracy

Figure 2 represents the detection accuracy of the proposed algorithm according to feature combination. In Table 1 the combination number denotes which three features were selected for that particular type combination. The proposed algorithm achieved the highest detection accuracy recorded as 98.76% when combination number 5 were selected (Please see table 1). This combination contains three important features

such as time to live value, TCP advertisement value, and the number of connections that contain same service and source address in previous 100 connections The lowest detection accuracy among the given results were recorded as 49.49% when combination number 4 were selected. So feature selection works as a contributing factor in increasing the intrusion detection accuracy.

Furthermore, the performance matrix of the proposed algorithm were analyzed. The false positive refers to a situation that there is an intrusion in the system but in reality it's not an intrusion. In figure 3 the percentage of false positive and in figure 4 the percentage of a false negative is shown for the proposed scheme.

The lowest false positive and false negative were recorded as 0.09% and 1.15% respectively while combination number 5 were taken into account. So it can be inferred that while the correlative features are selected the false positive rate decreases and it represents a reliable intrusion detection system.

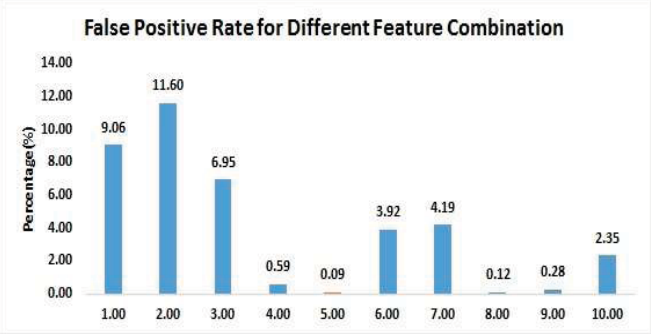


Figure 3: Percentage of False positive rate

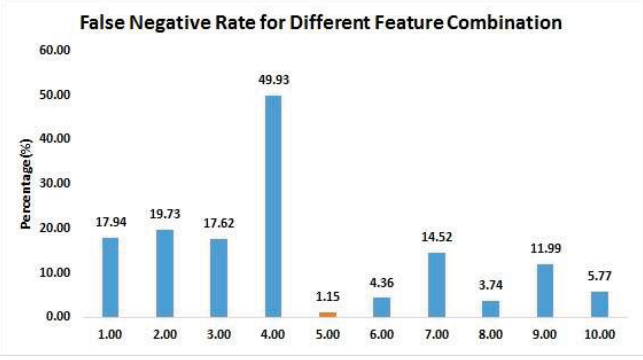


Figure 4: Percentage of False negative rate

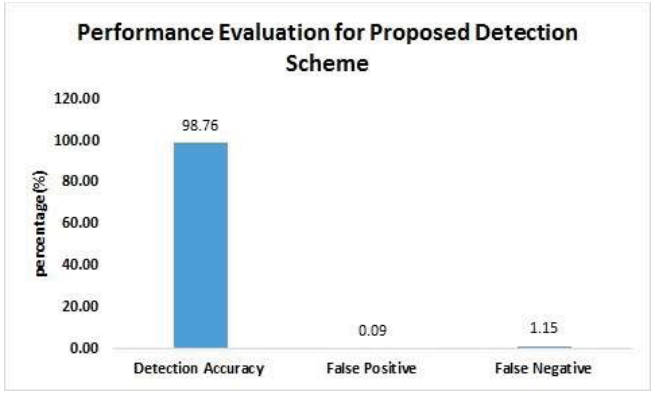


Figure 5: Performance Evaluation of the Scheme

Figure 5 represents the performance of the proposed scheme. The combination of time to live value, TCP advertisement value and the number of connections that contain same service and source address in previous 100 connections provided the highest detection accuracy with a very low false positive and false negative rate. Also using this feature combination the algorithm took only 13.84 seconds to detect an anomaly in the network traffic where the normal SVM method took 220.24 seconds to detect any anomalous behavior in the network.

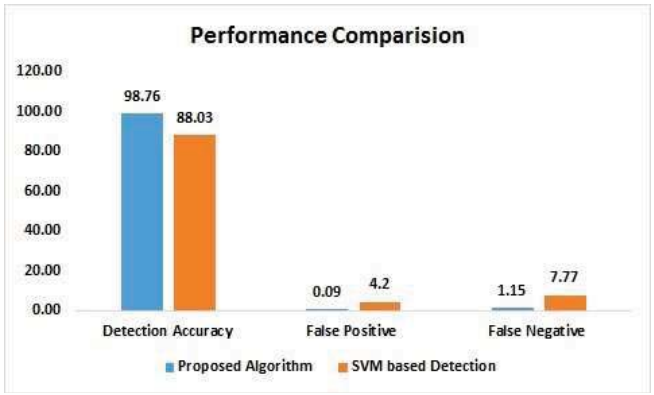


Figure 6: Performance Comparison

Figure 6 represents the comparison between the proposed algorithm and general SVM based detection scheme. As discussed before, in our algorithm we applied Simulated Annealing first to generate a combination of randomly selected 3 features and then SVM were applied. The normal SVM based scheme shows a detection accuracy of 88.03% only but it's hard to define which features needs to be taken into account to provide higher detection accuracy. Our proposed scheme provides 98.76% anomaly detection accuracy with lower false positive and false negative rate using

randomly three features only.

In this research, a set of three randomly selected feature were used to evaluate the performance. In future work, a different number of randomly selected features will be taken into account to evaluate the performance of the proposed algorithm.

6. CONCLUSIONS

In this paper, a combination of two machine learning method was used for network intrusion detection. The proposed algorithm provided significant detection accuracy of 98.76% and lower false positive rate of 0.09% and false negative rate of 1.15%, whereas the normal SVM based scheme achieved a detection accuracy of 88.03% and false positive rate of 4.2% and false negative rate of 7.77%.One of the important matter is the feature selection on which the most portion of the detection accuracy depends. Further research work can be done using a combination of a very low number of features that can reduce the time to detect an anomaly in the network traffic. Furthermore, Artificial Neural network will be applied to the dataset to evaluate the performance and compare with the proposed detection algorithm.

REFERENCES

[1] D. Batchelder, J. Blackbird, P. Henry, and G. MacDonald, "Microsoft Security Intelligence Report - Volume 17," *Microsoft Secur. Intell. Rep.*, vol. 16, pp. 1–19, 2014.

[2] K. Wang and S. Stolfo, "Anomalous payload-based network intrusion detection," *Recent Adv. Intrusion Detect.*, pp. 203–222, 2004.

[3] G. Xiaoqing, G. Hebin, and C. Luyi, "Network intrusion detection method based on Agent and SVM," *2010 2nd IEEE Int. Conf. Inf. Manag. Eng.*, pp. 399–402, 2010.

[4] R. P. Lippmann and R. K. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Comput. Networks*, vol. 34, pp. 597–603, 2000.

[5] E. Denning, R. Ave, and M. Park, "Attempted break-in --," pp. 118–131.

[6] W. Hu, Y. Liao, and V. R. Vemuri, "Robust anomaly detection using support vector machines," *Proc. Int. Conf. Mach. Learn.*, pp. 282–289, 2003.

[7] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," *Proc. IEEE Work. Inf. Assur. Secur.*, pp. 85–90, 2001.

[8] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," *Adv. Kernel MethodsSupport Vector Learn.*, vol. 208, pp. 1–21, 1998.

[9] Y. LeCun, L. D.Jackel, L. Bottou, A. Brunot, C. Cortes, J.~S.~Denker, H.~Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition," *Proc. {ICANN'95} - -International Conf. Artif. Neural Networks*, vol. II, pp. 53–60, 1995.

[10] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 130–136, 1997.

[11] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," *Comput. Vis. Pattern Recognition, 1997. Proceedings., 1997 IEEE Comput. Soc. Conf.*, pp. 193–199, 1997.

[12] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)."Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[13] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective (2016): 1-1 4.

Spectral Fractal Dimension Trajectory to Measure Cognitive Complexity of Malicious DNS Traffic

Muhammad Salman Khan, Sana Siddiqui, Ken Ferens, and Witold Kinsner
Dept. of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada
muhammadsalman.khan@umanitoba.ca, siddiqu5@myumanitoba.ca, ken.ferens@umanitoba.ca, witold.kinsner@umanitoba.ca

Abstract—Internet traffic exhibits long range dependence (persistence), scale invariance and self-similarity or self-affinity which are the known characteristics of fractals. Moreover, these characteristics of fractals can be extracted and quantified from an internet data time series using non-integer dimensions (fractal dimensions). The notion of cognitive complexity is also very well represented by the fractal dimensions, e.g., high value of fractal dimension of an object implies that the complexity of this object is higher than the one with lower fractal dimension. In addition, a multifractal object is more complex than a monofractal object and this can also be characterized to identify the degree of complexity. In this work, we have shown that the complexity introduced by distributed denial of service (DDoS) attack packets in DNS (Domain Name System) traffic is higher than the complexity of DNS traffic with no DDoS attack packets. A power spectrum density of the data series was used to calculate the spectral fractal dimension, and the performance of the proposed algorithm is validated using mathematical fractal Brownian motion process (fBm) and the real data sets. A sequence of spectral fractal dimension measurements of the time series (also known as a trajectory of spectral fractal dimension measurements or spectral fractal dimension trajectory (SFDT)) was generated to show the changing complexity of the series in time domain.

Keywords—Denial of service, Domain Name System (DNS), cyber threats, complexity, multifractal, power spectrum density, time series, spectral fractal dimension trajectory (SFDT), variance fractal dimension trajectory, malicious traffic.

I. Introduction

Certain cyber attackers exploit the vulnerabilities of DNS (Domain Name System) protocol to disrupt DNS services using various methods. Distributed denial of service (DDoS) DNS amplification attack is one of such methods which uses legitimate DNS servers to piggy back and amplify the payload of DNS packets. There is no useful information contained in such packets and they reduce the available bandwidth of the network. The attack is launched by sending a broadcast message to the legitimate computer nodes after manipulating the source and destination IP addresses of the message such that the receiver nodes receive these messages from an authentic node which acts as a piggy back node for the attacker. The victim nodes receive the DNS traffic continuously from the DNS servers without generating any DNS request [1] [2]. Since many authentic servers send these DNS response packets to the victim node, continuously, the resulting persistent high rate of traffic overwhelms the victim node. The victim node becomes unable to process the packets received at the rate they are being sent, and this causes the victim node to loose/drop packets, including packets received from other legitimate sources. Consequently, the victim node is unable to process other legitimate network requests, thus resulting in a denial of DNS service of those legitimate network requests. Furthermore, the attacker cannot be traced because the attack is launched using authentic source nodes and the attacker remains anonymous.

II. Literature Review

In order to detect DNS denial of service attacks with high accuracy, it is required to devise a solution that can differentiate accurately between normal and anomalous packet flows. Signature based methods cannot accurately detect DNS attacks, because there is no known signatures of DNS attack packets that can be used to differentiate between normal and attack packets. In other words, DNS attack packets resemble authentic DNS packets. However, there are various methods in the literature, which attempt to detect DNS DDoS amplification attacks. The authors in [2] describe a method of mapping and monitoring the DNS

mechanism of requests and responses to detect anomaly in the packet flows. This method shows better results in detection, but is limited due to the scaling issues in a large network. Moreover, it is useful for local DNS servers only. In [3], the authors utilized hardware based Bloom filters to analyze DNS packets to detect DNS amplification attacks. Also, as mentioned in [4] [5], there are location based and time based methods to detect DNS DDoS amplification attacks. There are various methods to detect these attacks and include packet based payload analysis and node based collaborative techniques. Fractal based estimation techniques are gaining popularity in anomaly detection algorithms due to their ability in looking at traffic patterns at multiple scales simultaneously. For example, authors in [6] proposed a correlation based fractal dimension for the detection of DDoS attacks using DARPA data set. They used a supervised learning mechanism to detect changes in fractal features. Cognitive security [7] and cognitive computing algorithms [8] [9] [10] [11] have shown their ability to mimic the apparent cognitive process that humans use to classify normal and anomalous traffic. In this work, the authors hypothesized that human's mental model to process information and classify normal and anomalous flows in a packet stream does involve an evaluation of the level of complexity of these flows. In the absence of attack signatures, cyber security experts utilize a cognitive model to differentiate between the complexities of a normal traffic flow from a malicious flow which results in further analysis of the malicious flow for a possible detection of a new threat. Malicious flows will have different level of complexity than that of legitimate traffic flow.

III. FRACTAL ANALYSIS

Typically, time series or data streams are analyzed using single(mono) scale analysis where any analysis i.e. statistical, spectral and/or transformation is performed on the data time series with equal sampling intervals. Multiscale analysis refers to analyzing the data series on multiple level of resolutions such that same data series is analyzed multiple times. Wavelet analysis is an example where independent time and frequency analysis is done that is equivalent to studying the time series at multiple resolution scales. Fractal based multiscale analysis is a revolutionary idea [12] [13] where a relationship is found between multiple resolution levels known as fractal dimension. This is akin to multiscale analysis simultaneously (and not independently as in wavelets) and finding how these multiscale levels are related which is equivalent of finding self-similarity. Multifractal analysis extracts the nature of fractality (fractional, or singularity, or non-integer behavior) of the object i.e. data time series. Mono-scale analysis is appropriate for any time series but cannot describe the relationship among various level of resolutions or scales. If the time series is self-affine then single scale analysis is not sufficient and multifractal analysis is required to extract the features (relationship among scales) [14]. In the science of cognitive detection, this is equivalent of measuring the complexity of the time series [15]. If the series is not very complex, then the value of fractal dimension is low and/or it may show monofractal behavior, while high values of fractal dimensions represent increasing level of complexity and/or the time series will have multifractal behavior. Fractal dimensions are always bounded by an upper value known as embedded dimension. Value of the embedding dimension represents the number of integer dimension of a time series. For example, for a single dimension time series (i.e. time series representing only one parameter such as count series of a variable), fractal dimension is bounded between 1 and 2 [14] [16].

IV. SPECTRAL FRACTAL DIMENSION TRAJECTORY

Spectral fractal dimension analysis, which is an extension of variance fractal dimension analysis [14], is a class of statistical/information based fractal dimension analysis where second order frequency analysis using power spectral density is performed at multiple scales and a relationship among those scales is found simultaneously using log-log relationship of multiple scales [14] [17]. Spectral fractal dimension analysis provides fractal dimension within the embedding topological dimensions of an object. For example, in this work, DNS packet count time series is a single dimension (1D) time series of packet count and therefore, the lower limit of topological dimension is 1 and upper limit is 2. For a 1D time series, spectral fractal dimension of 2 represents that the time series is generated from a statistical pink noise process while spectral fractal dimension of 1 represents a black noise process [18].

If a time series is a self-similar (or self-affine) fractal, the power spectrum density satisfies the following power law [17] [18] [19]:

$$\mathrm{P(f,T)} \sim (\frac{1}{f})^{d} \quad (1)$$

$\mathrm{P(f,T)}$ represents the power spectrum density of the time series as a function of the frequency and the window time T of the time series over which power spectrum density is calculated. Exponent d represents the slope of the least square fit of the line over power spectrum density plot. As shown in Figure 1, a line of slope 1 represents a

negative dimension over a PSD plot. This happens because, one sided PSD plot is considered to estimate the best least square fit which is a negative slope line. Therefore, we are required to reverse the sign in our calculations to ensure that dimensions remain positive. It is equivalent of considering the single sided negative frequency spectrum

As shown in Figure 1, lines having varying slopes over a log-log plot of PSD are shown [13] [14]. If the exponent of the equation (1) is −1, then the resulting PSD would be of blue noise where higher frequency components are amplified. Similarly, if d=0, then the resulting PSD would result due to white noise. If the frequency exponent is 1 then it represents PSD of a pink noise. For d=2, PSD is generated from brown noise or standard Brownian motion process. For d=3, it becomes black noise. Also increasing d from 1 till 3 will result in increasing attenuation of higher frequency components and the correlation will increase. Black noise is also called as broadband noise.

Moreover, this noise phenomenon is also called integer noise [18]. There are fractional noises that are not integer and lie between these integer limits. For example, if the exponent lies between 1 and 3, it is called fractional Brownian motion process [18]. As an example, Figure 2 shows a PSD plot of a Gaussian pulse while Figure 3 shows a one-sided plot of the same PSD plot. In order to find the spectral fractal dimension, Figure 4 shows a linear fit of the log-log plot of single sided PSD. Slope of this line is the magnitude of the exponent of equation 1. As there is only a single slope of the log-log plot of single sided PSD, this process is called a mono-fractal.

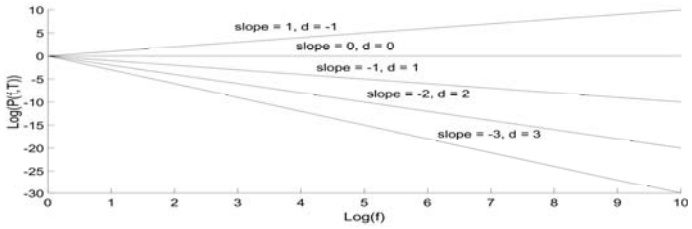


Figure 1: Slopes representing spectral fractal dimensions.

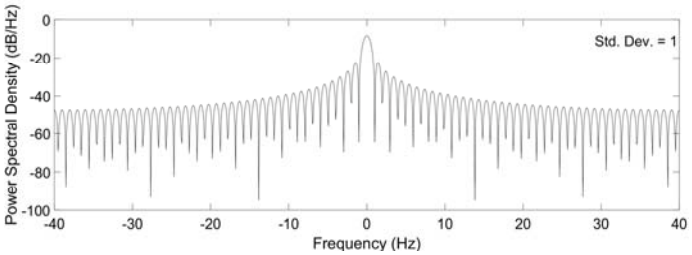


Figure 2: A double sided Power Spectral Density of a Gaussian pulse.

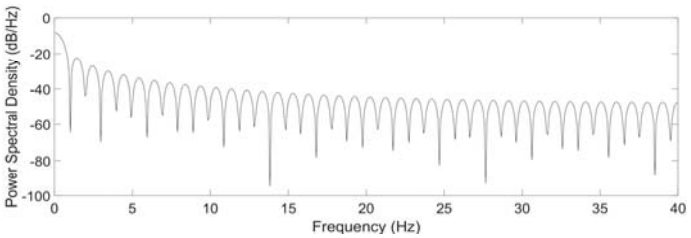


Figure 3: A single sided Power Spectral Density of a Gaussian pulse.

If there are more than one slope then the process is called multifractal and in this case we have to set the data window size T such that increasing the window size shows correct changes in the slope of the log-log plot of the PSD of the window. Therefore, if we estimate the spectral fractal dimension of a time series in a sliding window fashion, it will generate a multifractal trajectory that will represent the pattern of changing fractal dimension within the upper and lower limits of the topological dimensions [18].

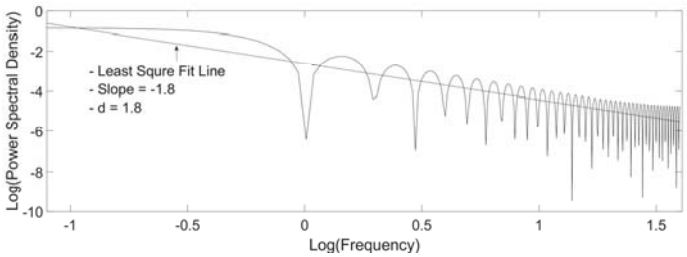


Figure 4: Log-Log plot and least square fit.

In order to find the spectral fractal dimension, following is the relationship between slope and the fractal dimension [13] [17]:

$$D_s = E + \frac{3 - d}{2} \tag{2}$$

where E is the number of dimensions or number of features represented by the time series. In this work, E =1, since there is one feature of the time series i.e. DNS packet count. Therefore, equation (2) is reduced to:

$$D_s = \frac{5 - d}{2} \tag{3}$$

Therefore, in this work, if spectral exponent is in the range $1 < d < 3$, then the spectral fractal dimension accordingly falls in the range $1 < D_s < 2$.

This work is a continuation of our ongoing research to explore and characterize complexity of internet data sets. Earlier [16] [20], authors have illustrated an algorithm using variance fractal dimension trajectory (VFDT) to characterize DNS time series using a moving window of data samples that is varying to ensure weak sense stochastic stationarity. In this work, an algorithm of

spectral fractal dimension trajectory (SFDT) is developed and tested on various DNS data sets to characterize the complexity of normal and attack DNS traffic. Main advantage of using spectral fractal dimension is that it does not require data samples to render weak sense stationarity which is a necessary requirement for variance fractal dimension trajectory. Spectral fractal dimension is an information based fractal dimension which is considered a frequency transform of variance fractal dimension and therefore, their theoretical computation results are bound to be same [21]. However, within a computational accuracy, this equivalence is not apparent. But as the results of this paper reveal, spectral fractal dimension is able to differentiate normal and attack traffic which is similar in performance of the variance fractal dimension. Moreover, in this work, 2 new data sets of normal DNS traffic are also used to test the performance of the proposed algorithm.

V. ESTIMATION OF POWER SPECTRAL DENSITY

There are 2 methods of estimating PSD of a time series [22]; non parametric methods and parametric methods. Non parametric methods estimate the PSD from the data itself using Fast Fourier Transform (FFT) and overlapping the adjacent windows. These methods introduce redundancy in the statistical information. Parametric methods are superior in performance because these methods seek to estimate the parameters of a linear or nonlinear model that is generating the time series. Typically these models mimic the model using statistical white process. Parametric methods are sometimes referred as Auto-Regressive (AR) processes whose order defines the type of non-linearity expected in the time series. In this work, authors have implemented a second order AR processes known as Yule-Walker method [23]. A time series can be represented as follows:

$$y[n] = a_0 x[n] + a_1 x[n-1] + \cdots a_N x[n-N] \qquad (4)$$

where $y[n]$ represents the time series of the window having an order of N. Parameters $[a_0, a_1, a_2 \ldots a_N]$ are required to be calculated to estimate the model generating the samples.

According to Yule-Walker model of estimating PSD of a time series, following is the calculation method [22]:

1. Set the model order N a-priori.
2. Find the autocorrelation function (ACF) of the N ordered AR process.
3. Find the FFT of the ACF which will provide an N'th order PSD estimate of the time series.

VI. DATA SET AND PROGRAMMING PLATFORM

Authors have used 3 data sets of DNS packets to analyze the performance of the proposed algorithm. Following is the summary of information about these data sets:

1. Data set from PREDICT USA [24] which contains traces of a DNS distributed denial of service attack (DDOS). This data set is composed of various packet capture (ERF file format) files taken from a real DNS attack scenario and is anonymized to ensure data confidentiality. It contains total 59,928,920 packets out of which there are total 358,019 DNS packets. DNS denial of service amplification attack was recorded for 10 minutes while the total capture time was 32 minutes and 47 seconds. One target IP and 6 DNS server IPs were already known and total 340,865 DNS denial of service packets were recorded. Total ERF file size is 5.3 GB. According to the USC-Lander [24], this data set was composed of one DNS Denial of Service Amplification attack staged between USC/ISI, Marina-del -Rey, Los Angeles, California to Colorado State University, Fort Collins, Colorado.
2. Data set from CAIDA USA [25] which contains internet traces from optical fiber internet connectivity from 2002 and 2003. Traces of April 24 2003 are used which captures 75,74,005 packets from 7:00:00 GMT till 7:04:59 GMT and contains 32,358 DNS packets within this duration. This does not contain any malicious traffic.
3. Data set from our experiment in which a PCAP file is captured from a lab computer which is being used for browsing and software development for the cyber security project. This computer is connected to internet and has MS Windows 7 installed. Firefox browsers are used for browsing and multiple windows and tabs are opened where many websites, cloud applications and services are connected to the internet. Approximate memory usage of Firefox based internet connectivity is 1.2 GB. 10,39,460 packets were captured on Nov 27 2015 from 00:08:06 GMT till 02:07:31 GMT. Out of these, total 11,721 DNS queries were made from 00:08:07 GMT till 02:07:21 GMT. This computer is heavily guarded against any malicious threats by the network administrators.

The proposed algorithm is developed using Matlab programming platform. DNS data time series is created using Matlab based parsing program such that the time series represents DNS time series at individual end points

of a network. DNS data time series is generated by sampling DNS packets at 100 milliseconds in order to include the network latency to complete a packet round trip is covered sufficiently [26].

VII. ALGORITHM

Spectral Fractal Dimension Trajectory (SFDT)

1. Set the following parameters:
   a. Data pointer: d_p.
   b. Window size: lag. (use 1024 samples for DNS packets which is less than 2 minutes of window for 100 ms sampling and is aligned with DNS traffic patterns i.e. normal DNS traffic is not very frequent compare to HTTP or FTP traffic)
   c. Window= d_p+ lag.
   d. Auto-Regression process degree: d=2.
   e. Feature Dimension: E=1.
2. Initialize $\boldsymbol{d_p}$ at first sample of the data series.
3. Run a main loop till the end of data series.
4. Pass the window $N = d_p + lag$.
5. Set M-point FFT:
$$MFFT = 2^{\text{ceil}(\log_2(N))}$$
6. Call **PSD_AR(N, *MFFT*,d)** function and get one-sided estimate of power spectral density (PSD).
7. Take logarithm of both PSD and frequency.
8. Estimate the least square slope of the log-log plot of PSD.
9. If slope is greater than -1, then remove high frequency components of PSD till slope $<$ -1.
10. If slope is less than -3, then remove low frequency components of PSD till slope $>$ -3.
11. If slope is not defined due to computation limitations or zeros in the data set (for which log is undefined), use previous value of slope because having a zero slope or log of zero means that the power spectral density is either a flat line which means it is a white process or there is a zero value whose fractal dimension is zero.
12. Change the sign of slope and calculate spectral fractal dimension according to equation (3).

Power Spectral Density Estimate Function- PSD_AR( )

1. Let N samples are considered in a window of data X: *main window* X_N.
2. Remove DC component as follows:
$$X_C = X_N - \frac{sum(X_N)}{N}$$
3. Calculate sampling frequency:
$$F_s = \frac{N-1}{\max(X_N)}$$
4. Append 0's at the end of X_N corresponding to the difference: $N - MFFT$.
5. Estimate parameters a_0, a_1, a_2 of the AR(2) process using Yule-Walker model. Find the autocorrelation vector ACF_x of the $MFFT$ samples.
6. Calculate the Fast Fourier Transform of the ACF_x that will output a vector of the least square estimate of the power spectral density (PSD) vector of the time seriesX_N.
7. If ACF_x has even elements, return following number of elements of estimated PSD vector:
$$\frac{MFFT}{2} + 1$$
Else return following number of elements of estimated PSD vector:
$$\frac{MFFT + 1}{2}$$

VIII. EXPERIMENT AND RESULTS

In order to validate the performance of spectral fractal dimension trajectory, authors have tested the algorithm over fractal Brownian motion (fBm) process. We have generated different samples of fBm process using varying values of Hurst parameter between 0.1 and 0.9. As spectral fractal dimension trajectory follows equation 3 for a single feature, therefore, spectral fractal dimension trajectory is obtained that varies between 1.9 and 1.1 respectively for the above values of Hurst parameters. As shown in Figure 5, 30780 samples of a concatenated fBm process are shown which are generated with 3 values of Hurst parameter; 10260 samples for H=0.1, next 10260 samples for H=0.5 and the rest with H=0.9. As can be seen in Figure 6, SFDT shows marked variations in the trajectory when Ds changes. Spectral fractal dimension trajectory of this concatenated fBm process clearly shows that SFDT for the 3 different fBm processes is distinctly apparent and the first and last samples of each fractal process are following closely with the process itself.

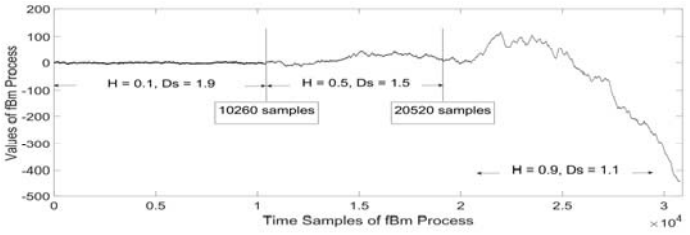


Figure 5: Concatenated fBm process with varying Ds.

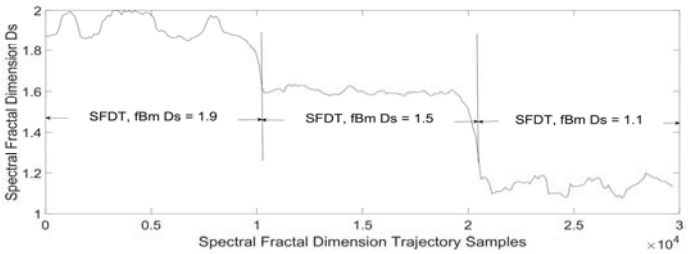


Figure 6: SFDT of Concatenated fBm process.

In this work, we have tested the spectral fractal dimension trajectory algorithm to characterize the time series of DNS packet counts using data set from CAIDA [25], PREDICT [24] and our own experimental data set.

As Figure 7 shows, a sampled time series of DNS packet counts from CAIDA data set is generated. We can see that it shows many samples with no DNS packets. In addition, there is only one sample that has maximum count of 30 while most non-zero samples lies close to the count of 10. As evident from Figure 8, the spectral fractal dimension trajectory of this time series is showing a constant fractal dimension within a precision order of 13 decimal digits which can be rounded to a fractal dimension of 1.99. This is a monofractal and as already discussed in Figure 1, this behavior can be approximated to a random fractional Brownian motion process which in turn is equivalent to a Brown noise process. Therefore, the time series of DNS packet count would have strong correlation which is a property of Brown noise process. In order to ensure and further validate that normal DNS packet counts are monfractal, we have tested the algorithm over an experimental data set of normal DNS packets whose DNS count time series sampled at 100 millisecond is shown in Figure 9. This series conforms the pattern of CAIDA data set but the max DNS packet count is 12 which is lower than CAIDA data set. It is expected since CAIDA data set is generated by monitoring traffic at the network gateway of an optical fiber link while this experiment is performed over a single computer in our lab. However, most samples are found below a count of 6 and there are many samples with zero DNS packet counts. As depicted in Figure 10, spectral fractal dimension trajectory is showing monfractality within the range of 1.99. Again, it confirms our hypothesis that normal DNS packets are fractal Brownian process and does not show higher degree of complexity as multifractals do.

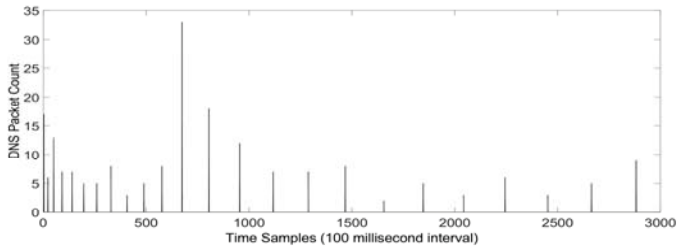


Figure 7: Time series of DNS packet counts sampled at 100 ms – CAIDA data set.

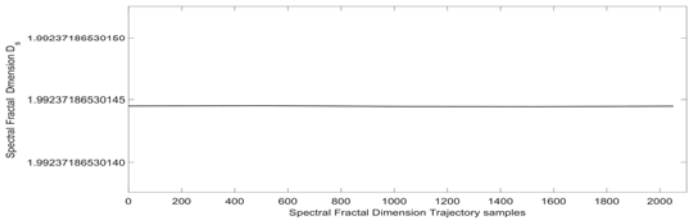


Figure 8: SFDT of DNS packet counts – CAIDA data set.

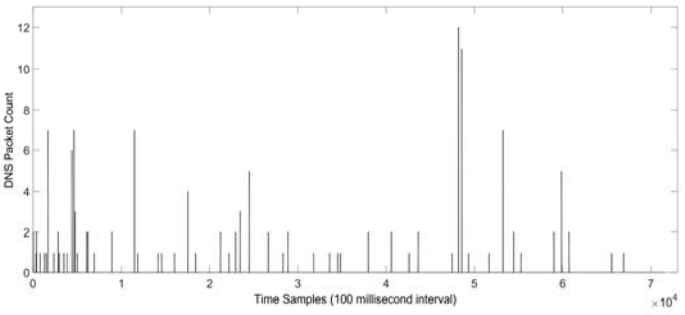


Figure 9: Time series of DNS packet counts sampled at 100 ms – Experiment data set.

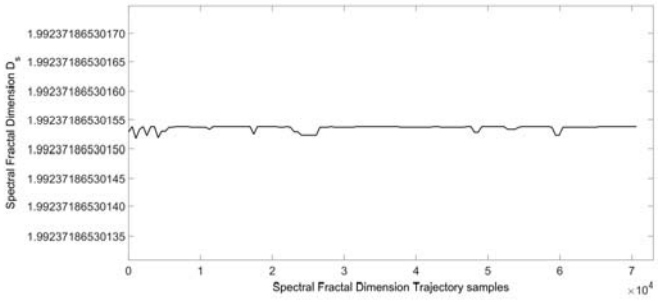


Figure 10: SFDT of DNS packet counts – Experiment data set.

As shown in Figure 11, DNS packet count time series sampled at 100 millisecond is shown for PREDICT data set. It contains both normal and attack traffic. SFDT algorithm is applied and Figure 12 shows the trajectory of spectral fractal dimension of DNS time series. This time series of DNS packet counts over the target IP is generated with equal sampling interval of 100 millisecond. Attack started at sample number 805 and ended at sample number 9146. There are total 10260 samples. Moreover, at sample number 72, there is a large burst of DNS packet count that

happens when the node starts sending and receiving DNS broadcast to resolve queries and build local DNS cache etc. As can be seen, attack starts when the spectral fractal dimension is above 1.8 and ends when it goes below 1.8. During the start and end time, spectral fractal dimension trjectory shows higher dimension close to 1.9. These values of start and end of the attack are dependent on data set. However, we can also state that the presence of attack has introduced higher level of complexity as depicted by the change (increase) in spectral fractal dimension i.e. multifractal. Moreover, there is a significant distinction between normal and attack DNS traffic. Correspondingly, the attack traffic is not showing a monofractal behavior and has varying fractal dimension between the range of 1.8 and 1.9.

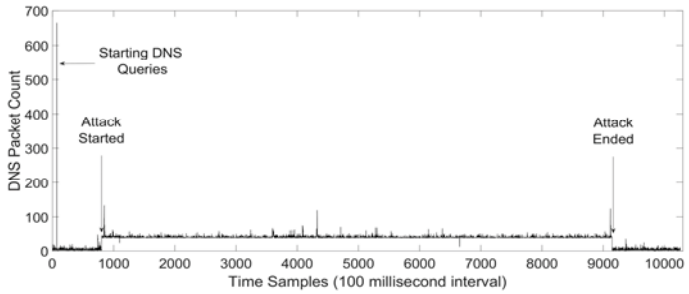


Figure 11: Time series of DNS packet counts sampled at 100ms – PREDICT data set.

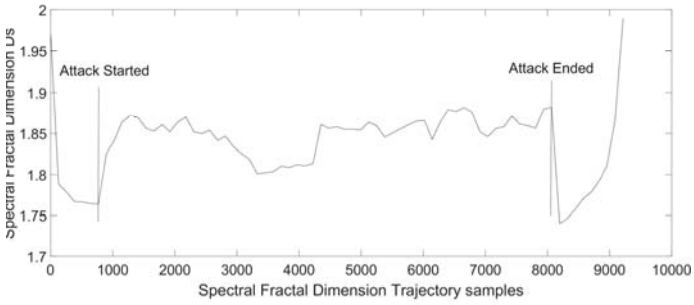


Figure 12: SFDT of DNS packet counts containing attack – PREDICT data set

IX. DISCUSSION

The spectral fractal dimension trajectory (SFDT) is a proposed method that calculates the cognitive complexity of a time series in a sliding window fashion by estimating the self-similarity or self-affinity of the sliding window of the time series. As shown, a mono-fractal such as a single dimension fractal Brownian motion process shows a mean value of Ds which is a least-square estimate of the spectral fractal dimension. Moreover, multifractal which shows varying spectral dimension over different time intervals (time windows), exhibits significant variations in fractal dimension trajectory (SFDT) as the self-similarity or fractal dimension changes with the course of time. Equivalently, multifractals show a varying degree of persistence (contrast to monofractal which have single degree of persistence) from high degree of persistence at one extreme to high degree of anti-persistence at other extreme. It can be stated that multifractals are more complex because they have varying fractal dimensions which is equivalent to varying degree of persistence. It is important to note that spectral fractal dimension trajectory is sensitive to errors in calculation due to outliers and saturation points. As spectral fractal dimension is calculated by estimating the slope of a least-square fit of line over a log-log plot of the single-sided power spectral density of time series, it is critical to remove the saturation points and outliers in the data. If the dimension is going below the lower limit of the topological dimension i.e. E =1, then remove low frequency components which introduce saturation. Moreover, if the dimension is going above the higher limit of the topological dimension i.e. E=2, then remove the higher frequency components which introduce outliers and increases the slope of the PSD. Moreover, if a window of time series contains zero values, then it is an indication of no fractality. Also, if there is a zero slope in the power spectral density, then it should be treated as random process.

In addition, it is considerably important to pre-process the packets carefully. In our work, we have provided an initial proof of concept to detect an attack by estimating the spectral fractal dimension in a sliding window fashion. We call this spectral fractal dimension trajectory (SFDT). However, SFDT algorithm requires that the time series should follow Nyquist sampling criterion to generate statistically valid samples. This requires that the standard characteristics of the observation data are known a-priori i.e. round trip time of the DNS packet, so that the time series represents meaningful features i.e. DNS packet count in our work.

As stated in the experiment and result section, our preliminary analysis shows that normal DNS packet count time series shows high degree of mono-fractality which is a sign of low complexity as compared with multifractal time series where normal and attack traffic shows different fractal behavior (or fractal dimensions). It is also shown that if there is a consistent DDoS DNS attack, then the multifractal dimensions will show lot of complexity/fractal variations especially during the time of start and end of an attack. It can be validated through visual plots and could provide an automated way to alert the network administrators.

It can be argued that a DNS DDoS attack can be detected visually then why there is a need to automate the detection process. It is imperative to note here that with the state of the art security technologies that includes firewall, Security Information and Event Management Systems (SIEM) and Intrusion Detection Systems (IDS), human analysis is required to either analyze and/or configure the statistical results of the internet flows. Moreover, main purpose of DDoS attack is to deny availability of a network service to a network or a node in a network. As most of the security systems are based on perimeter security (network security), it is hard to detect a DDoS attack if it is aimed at a particular node because for the network security administrators, detecting an anomalous increase in packet counts for a node would be equivalent of finding a needle in the haystack of humongous network data to detect. In addition, machine learning can be used to detect DDoS attacks autonomously, but as shown in Figure 7 and Figure 9, normal network data does not show consistent pattern that is necessary to benchmark machine learning algorithms based detection of DDoS attacks. As authors have already shown in [27] , probability of false alarms is relatively high and lot of fine tuning and reconfiguration of machine learning algorithm is necessary. Therefore, in this work, it is presented that the detection of DDoS attacks can be offloaded to a cognitive algorithm which is based on spectral fractal dimension trajectory.

X. CONCLUSION

In this work, authors have presented a new fractal based cognitive algorithm called spectral fractal dimension trajectory (SFDT) to detect variations in the complexity of the DNS packet time series using a sliding window. SFDT generates statistically valid spectral fractal dimensions over a sliding window of time series and the power spectrum density of the sliding window is estimated using second order auto-regression process. Also, authors have validated the performance of the algorithm using mathematical fractal Brownian motion process. The proposed algorithm is prone to the high variability of the time series and can capture variations in the complexity of the time series due to the presence of an attack. Also, it is shown that normal DNS traffic either at a network gateway or at a node in a network shows a monofractal behavior with persistent fractal dimension while in the case of DNS DDoS attack, spectral fractal dimension trajectory shows multifractal behavior which is an indication of an increase in degree of complexity from monofractal to multifractal.

XI. ACKNOWLEDGMENT

This work is supported in part through a research fellowship from Mitacs-Accelerate Canada. Authors are also thankful to PREDICT USA and CAIDA USA for providing state-of-the-art data sets.

XII. REFERENCES

[1] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window," in *Proceedings of IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2015*, Beijing, China, 2015.

[2] Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis and Stefanos Gritzalis, "Detecting DNS amplification attacks," *Lecture Notes in Computer Science,* vol. 5141, pp. 185-196, 2008.

[3] Changhua Sun, Bin Liu and Lei Shi, "Efficient and low-cost hardware defense against DNS amplification attacks," in *IEEE GLOBECOM*, 2008.

[4] Saman Taghavi Zargar, James Joshi and David Tipper,, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," in *IEEE Communications Surveys & Tutorials*, 2013.

[5] Tao Peng, Christopher Leckie and Kotagiri Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *Journal ACM Computing Surveys (CSUR),* vol. 39, no. 1, 2007.

[6] Zhengmin Xia, Songnian Lu and Jianhua Li, "DDoS Flood Attack Detection Based on Fractal Parameters," in *Proceedings of 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Shanghai, China , 2012.

[7] Witold Kinsner , "Towards cognitive security systems," in *Proc. 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, Kyoto, Japan; August 22-24, 2012, 2012.

[8] Yingxu Wang, "On cognitive informatics," in *Proc. 1st IEEE Intern. Conf. Cognitive Informatics*, Calgary, 2002.

[9] Simon Haykin , Cognitive dynamic systems: Perception-Action cycle, Cambridge, UK: Cambridge University Press, 2012, p. 322.

[10] Penti O.A. Haikonen , The cognitive approach to consious machines, New York, NY: Academic, 2003.

[11] Yingxu Wang, Du Zhang and Witold Kinsner, Advances in cognitive informatics and cognitive computing, vol. SCI 323, Berlin: Springer Verlag, 2010, pp. 265-295.

[12] Witold Kinsner , "It's time for multiscale analysis and synthesis in cognitive systems," in *Proc. IEEE 10th Intl. Conf. Cognitive Informatics & Cognitive Computing (ICCI*CC11)*, Banff, AB, 2011.

[13] Michael Potter and Witold Kinsner, "Multifractal characterization of synthetic ECG in the presence of coloured noise," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, 2004.

[14] Witold Kinsner, "A unified approach to fractal dimensions," *Int'l Journal of Cognitive Informatics and Natural Intelligence,* vol. 1, no. 4, pp. 26-46, 2007.

[15] Witold Kinsner, "Towards cognitive machines: Multiscale measures and analysis," *Intern. J. Cognitive Informatics and,* vol. 1, no. 1, p. 28–38, 2007.

[16] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window," in *Proc. IEEE 13th Intl. Conf. Cognitive Informatics & Cognitive Computing (ICCI*CC14)*, 2015.

[17] Joao B. Florindo and Odemir M. Bruno , "Fourier fractal descriptors for colored texture analysis," in *Lecture Notes in Computer Science, Advanced Concepts for Intelligent Vision Systems*, 2011.

[18] Witold Kinsner, *Graduate lectures on Fractal and Chaos Engineering,* Winnipeg, MB, Canada, 2015.

[19] Joao B. Florindo and Odemir M. Bruno, "Closed contour fractal dimension estimation by the fourier transform," *Chaos, Solitons and Fractals,* vol. 44, no. 10, pp. 851--861, 2011.

[20] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A polyscale autonomous sliding window for cognitive machine classification of malicious Internet traffic," in *The 14th International Conference on Security and Management (SAM'15), WORLDCOMP'15*, Las Vegas, 2015.

[21] Witold Kinsner, "Towards Cognitive Machines: Multiscale Measures and Analysis," in *Novel Approaches in Cognitive Informatics and Natural Intelligence*, 2009, pp. 188-199.

[22] Inan Guler, M.Kemal Kiymik, Mehmet Akin and Ahmet Alkan, "AR spectral analysis of EEG signals by using maximum likelihood estimation," *Computers in Biology and Medicine,* vol. 31, no. 6, p. 441–450, 2001.

[23] Larry Marple, "A new autoregressive spectrum analysis algorithm," *IEEE Transactions on Acoustics, Specch and Singal Processing,* Vols. ASSP-28, pp. 441 - 454, 1980.

[24] PREDICT, "USC/Lander - Scrambled Internet Measurement, PREDICT ID USC-Lander".

[25] CAIDA UCSD , 2003. [Online]. Available: https://data.caida.org/datasets/oc48/oc48-original/.

[26] AT&T Inc. USA, "Network latency," 2015. [Online]. Available: https://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html. [Accessed 2015].

[27] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A chaotic complexity measure for cognitive machine classification of cyber-attacks on computer networks," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI),* vol. 8(3), 2014.

[28] Delio Brignoli, "A Masters thesis on "DDoS detection based on traffic self-similarity"," University of Canterbury Research Repository, 2008.

SESSION

CRYPTOGRAPHIC TECHNOLOGIES I

Chair(s)

Prof. Levent Ertaul

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

Secure Computing through Homomorphic Encryption on a Peer-Servicing Public Cloud Computing Platform

Sheng-Tzong Cheng and Yin-Jun Chen
Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan

Abstract*—Cloud computing is able to redistribute the computing resources, and then reduces the maintenance cost, thereby increases profits. Ensuring data security and maintaining confidentiality of personal information are important issues of cloud computing security. In order to avoid malicious attacks and data theft in the applications of cloud computing, homomorphic encryption provides a novel technique for computing with zero-knowledge privacy. Fully homomorphic encryption, proposed by Gentry in 2009, allowed the execution of all kinds of secured computation without the secret key. Data privacy in the public cloud could be achieved in this way.*

This study proposes a cloud-computing framework utilizing fully homomorphic computation for processing the customer's pre-encrypted data. The proposed framework is able to compute and to verify the result without the need of decrypting the data in the whole process. The integration of homomorphic scheme with garbled circuits provides a mechanism of verifiable computation. It avoids the security vulnerability where data is kept by an entrusted third party. In addition, this paper builds a system on a peer-servicing public cloud platform powered by the P2P Hadoop. Accordingly, there is no single point of failure in this high-availability system. The experiments show that the security of this system meets the standards. In conclusion, the proposed framework has confidentiality, integrity and availability complying with core criterions of information security.

Keywords: Cloud computing, MapReduce, Hadoop, Homomorphic Encryption, Information Security

1 Introduction

In the wake of cloud computing, how to keep the user's data confidential but still manageable for cloud service provider remains an important but unresolved issue. The existing security measure only creates encrypted communication channels between users and cloud services. However, the cloud service providers still possess the unencrypted user data. It affects the confidentiality of client's data. Therefore, we adopt homomorphic encryption to the data for cloud services without decrypting them.

In this paper, we focus on a homomorphic encryption framework on a high-availability cloud computing platform. By exploiting this framework, we are able to execute the computation of confidential data without decrypting. Confidentiality, integrity, and availability of information security are the core criterions.

1) *Confidentiality*: Confidentiality could be enforced by a set of rules to limit the access to the data and to assure that data is not disclosed by unauthorized persons, processes, or devices.
2) *Integrity*: Integrity is to guarantee the accuracy, security, and consistency of data over its entire life-cycle. It includes
   a) Accountability: Ability to trace which requirement is issued by which entity.
   b) Authenticity: It provides the sender's identification in a message, a file, a computer system, a software process, or even a database to users. It ensures and validates the data, transactions, communications, or documents to be genuine.
   c) *Non-repudiation*: It is to record actions and/or events which make actions and/or events not deniable after occurrence. It is known as digital signatures, public key encryption, etc.
   d) *Reliability*: It indicates the probability for a system to accurately execute its function in a period of time.
3) *Availability*: It refers to ensuring that authorized parties are able to access the information when needed.

As the growing demand of processing big data, cloud computing has become one of the most popular researching subjects in recent years. One of the major cloud computing frameworks is Hadoop. It provides an interface to implement MapReduce which allows people to use it more easily. As a result, Hadoop has been widely used in the distributed computing environment nowadays.

In a dynamic cloud environment, there are some problems about node churn and failures including master failure and it results from a large number of computing nodes joining and leaving the network at very high rates. Therefore, an effective mechanism to manage such problems is fundamental to make MapReduce applications reliable, while the current MapReduce middleware could be unreliable. Therefore, P2P-MapReduce is an adaptive MapReduce framework by us to solve the problem about node churn and master failures. It provides a decentralized but effective way for job recovery, so it can make MapReduce more reliable in dynamic Cloud infrastructures.

In this work, we consider Java Native Interface because it allows users to execute Homomorphic Encryption programs written in C++ with little modification to the HElib source code and less overhead. The system overview is shown in Figure 1. Upon receiving a job, master node assigns these tasks to workNodes, and it uses JNI to call HElib.

ISBN: 1-60132-445-6, CSREA Press ©

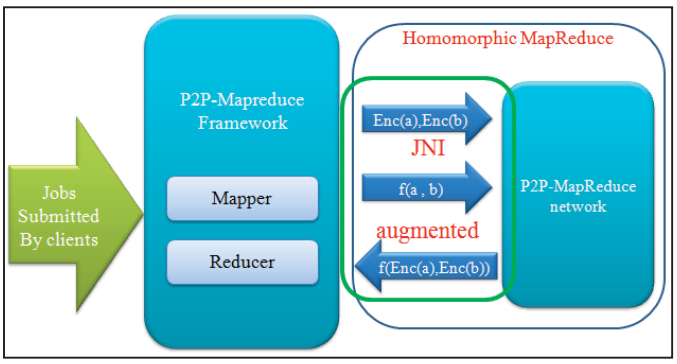


Fig. 1. System overview

2. Background

2.1 MapReduce

MapReduce is an application development framework on data centers with thousands of computing nodes. It is also a programming model that allows the development of applications capable of processing great amount of data on large clusters. Users don't need to handle parallelization, remote execution, data distribution, load balancing, and fault tolerance. There are several systems such as Google MapReduce [1], Hadoop [2], and Phoenix [3], implementing the MapReduce model for CPU-based clusters.

There are two user-defined functions in MapReduce applications, map function and reduce function. Developers can utilize interfaces to do a number of real-world jobs such as big data processing, machine learning, or image processing in massively parallel program models.

The map function uses a set of splitting input and generates a set of corresponding intermediate key-value pairs which are defined by the user. And these intermediate pairs are sorted by keys and shuffled to the reduce function.

The same keys with a set of values are transmitted to a reduce function. The user-defined reduce function merges together these values to generate the final result.

2.2 P2P-MapReduce

There are two main goals of the P2P-MapReduce framework: (i) to dynamically replicate the jobs on a set of backup master nodes to handle the problem of master node failure; and (ii) to allow the peer node to join and to leave the system at unpredictable rates. This framework is capable of supporting MapReduce program model in dynamic P2P networks.

In order to realize the P2P model, P2P-MapReduce is defined as an architecture where each node can be a role either as a master or slave. The role of a node can change dynamically in different time. Therefore, a limited set of nodes is configured as the master characteristics and the other nodes are configured as the slave characteristics.

It is worth of noticing that the mechanism to recover a failed task is transparent to the user. There is also a point to notice that a master node may play the role of primary master for one job and that of backup master for another job at the same time [15].

2.3 Homomorphic Encryption

Homomorphic encryption is a method of encryption which is able to compute encrypted data, and then to generate an encrypted result. When the result is decrypted, the decrypted result matches the result of operations performed on the plaintexts. The following description is the definition of Homomorphic encryption.

Definition: An encryption is homomorphic if a and b are encrypted separately as Enc(a) and Enc(b), and it is possible to compute Enc($f(a, b)$) without using the private key, where f can be +, ×, or $\oplus$.

We classified the kinds of Homomorphic encryption, according to the operations that are able to evaluate on original data. The additive Homomorphic encryption could be the Pailler [4] or Goldwasser-Micalli [5] cryptosystem. The multiplicative Homomorphic encryption is either RSA [6] or El Gamal [7] cryptosystem.

a) Addittive Homomorphic Encrption: Enc $(x \oplus y)$ = Enc(x) $\otimes$Enc(y),

In the Paillier cryptosystem, if the public key is the modulus m and the base is g, then the encryption of a message x is $\mathcal{E}(x) = g^x r^m \bmod m^2$, for some random $r \in \{0, \ldots, m\text{-}1\}$. The homomorphic property is $\mathcal{E}(x_1)\cdot\mathcal{E}(x_2) = (g^{x1}r_1^m)(g^{x2}r_2^m) = g^{x1+x2}(r_1r_2)^m = \mathcal{E}(x_1 + x_2 \bmod m^2)$.

b) Multiplicative Homomorphic Encryption: Enc $(x \otimes y)$ = Enc(x)$\otimes$Enc(y)

If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by $\mathcal{E}(x) = x^e \bmod m$. The homomorphic property is then $\mathcal{E}(x_1)\cdot\mathcal{E}(x_2) = x_1^e x_2^e \bmod m = (x_1x_2)^e \bmod m = \mathcal{E}(x_1 \cdot x_2)$.

The homomorphic library what we use is HElib, and it is based on the algorithm which is proposed by Halevi [8], Brakerski, and Vaikuntanathan(BGV) independently. They found very different ways to construct FHE without using the squashing step, and thus without the sparse subset sum assumption. These schemes are derived from Gentry's blueprint for homomorphic encryption. The security what Brakerski and Vaikuntanathan use is based on LWE for sub-exponential approximation factors, avoiding reliance on ideal lattices.

We now briefly introduce the description of learning with error (LWE). LWE is a problem in machine learning that is conjectured to be hard to solve. It is a generalization of the parity learning problem introduced by Oded Regev [9] in 2005. Regev also showed that the LWE problem is as hard to solve as several worst-case lattice problems. The LWE problem [9, 10] has recently been used as a hardness assumption to create public-key cryptosystems such as the ring learning with error key exchange by Peikert [11].

2.4 Yao's Garbled Circuit Construction

The following description is about Yao's protocol for two-party private computation [12, 13]. More details information about this could be found in the work by Lindell and Pinkas [14].

In general, we assume there are two parties, Alice and Bob. And they wish to compute a function F over their private inputs *a* and *b*. We focus on polynomial-time deterministic functions for making easy to understand the construction.

First, Alice needs to convert F into a boolean circuit C. She generates a garbled version of the circuit, called G(C), and sends it to Bob. The sending includes a garbled version of her input, called $G(P_0)$. Alice and Bob then engage in a series of oblivious transfers so that Bob can obtain a garbled version of his input, called $G(q_0)$, but Alice won't learn anything about *b*. Bob then calculates the garbled circuit with the two garbled inputs to derive a garbled version of the output, called $G(F(P_0,q_0))$. Alice can then translate this into the actual output and share the result with Bob. We assume this protocol follows an honest-but-curious adversary model.

3. System design

The details of the system design and enhanced model are elaborated in this chapter. First, the considered problem is formulated in Section 3.1 and then the objective function and the parameter definition are listed in the following section. In Section 3.2, we formulate our enhanced model and compare our model with other block cipher mode model and elaborate on the parameters. In Section 3.3, we describe a verifiable computation scheme which is combined with homomorphic encryption scheme with garbled circuit. In the last section, we detail our enhanced FHE model with pseudo code, flow chart, and mathematical analysis to illustrate how the scheme works.

3.1 Problem Description

The original homomorphic encryption scheme is inefficient, because it encrypts plaintexts per a char or an integer, and it decrypts once for each cipher text. To improve the performance, we propose a model to encrypt plaintexts once a block and to decrypt cipher text once a block.

The previous homomorphic encryption schemes are lack of data integrity. And, it is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. Thus, to provide a mechanism of checking integrity, we propose a cipher mode of operation in which each block of plaintext is XORed with the checksum of corresponding plaintext block before being encrypted.

3.2 Enhanced Cipher of FHE

In this section, we formulate our problems depicted earlier and elaborate on the parameters. We add checking data integrity mechanism and block cipher model into BGV scheme, more detail description is given in the following sections.

3.2.1 Checksum Block Cipher

A mode of operation is an algorithm that uses a block cipher to provide an information service such as confidentiality or authenticity in cryptography. A block cipher by itself is only suitable for the ciphering purpose. Cipher Block Chaining has been the most commonly used mode of operation. Its main drawbacks are that encryption is sequential and that the message must be padded to a multiple of the cipher block size. One way to handle this issue is tackled through the method known as ciphertext stealing. When one bit is changed in a plaintext or IV, it affects all following ciphertext blocks. Figure 2 gives the demonstration of cipher block chaining.

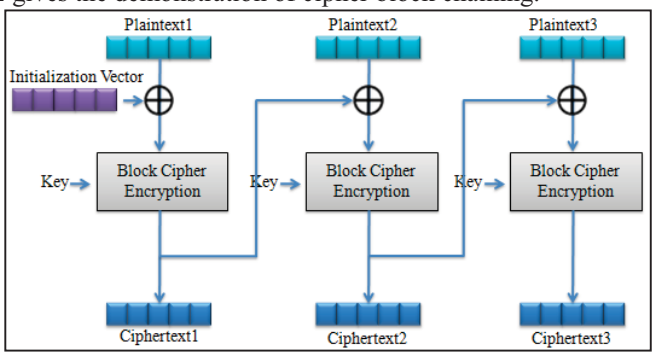


Fig. 2. Encryption of Cipher Block Chaining

Decrypting with the incorrect Initialization Vector (IV) causes the first block of plaintext to be corrupt but subsequent plaintext blocks are correct. This is because a plaintext block can be recovered from two adjacent blocks of ciphertext. As a consequence, decryption can be parallelized. When one bit is changed to the ciphertext, it causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext, but the rest of the blocks remain intact. This feature is exploited in different padding oracle attacks, such as POODLE. Figure 3 shows the demonstration of cipher block chaining.

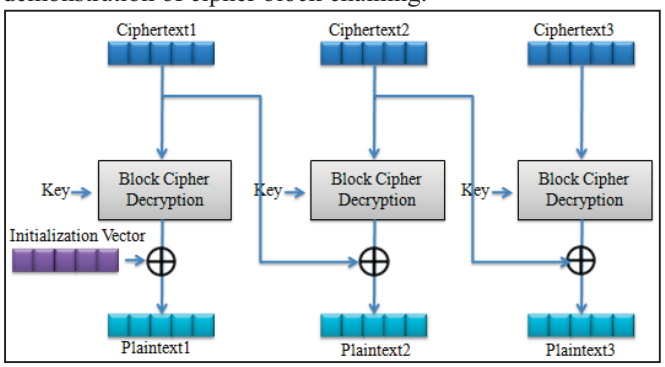


Fig. 3. Decryption of Cipher Block Chaining

Explicit IVs take advantage of this property by prepending a single random block to the plaintext. Encryption is done as normal, except the IV does not need to be communicated to the decryption routine. Whatever the decryption of IV uses, only the random block is corrupted. It can be safely discarded and the rest of the decryption is the original plaintext.

Confusion means that each character of the ciphertext should depend on several parts of the key. Diffusion means that if we change a character of the plaintext, then several characters of the ciphertext should change, and if we change a character of the ciphertext, then several characters of the plaintext should change [16].

In Shannon's original definitions, confusion refers to making the relationship between the ciphertext and the symmetric key as a complex relationship; diffusion refers to dissipating the statistical structure of plaintext over bulk of ciphertext. This complexity is generally implemented through a well-defined and repeatable series of substitutions and permutations. Substitution is referred as the replacement of certain

components with other components, following certain rules. Permutation is referred as the manipulation of the order of bits according to some algorithm. To be effective, any non-uniformity of plaintext bits needs to be redistributed across much larger structures in the ciphertext, making that non-uniformity much harder to detect. In particular, for a randomly chosen input, if one flips the *i*-th bit, then the probability that the *j*-th output bit changes should be one half, for any *i* and *j*—this is termed the strict avalanche criterion. More generally, one may require that flipping a fixed set of bits should change each output bit with the probability of 0.5.

The simplest way to achieve both diffusion and confusion is to use a substitution-permutation network. In these systems, the plaintext and the key often have a very similar role in producing the output, hence the same mechanism ensures both diffusion and confusion.

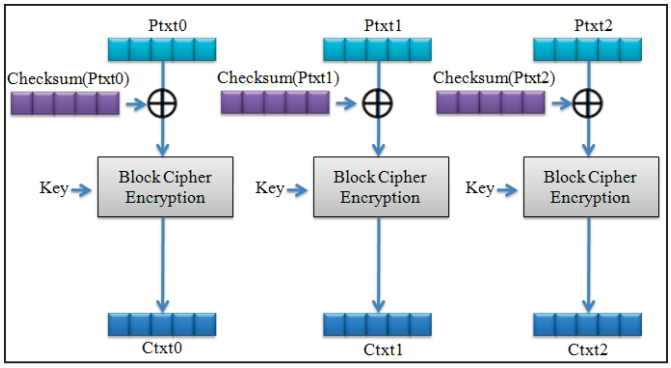


Fig. 4. Encryption of Checksum Block cipher

In Figure 4, if the first block has index 0, the mathematical formula for our checksum block cipher encryption is $C_i = E_k(P_i \oplus \text{Checksum}(P_i))$, while the mathematical formula for our block cipher decryption is $P_i = D_k(C_i) \oplus \text{Checksum}(P_i)$ in Figure 5. The checksum function is CRC32.

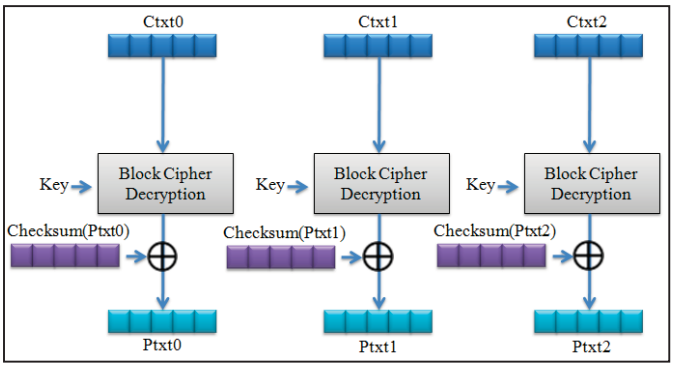


Fig. 5. Decryption of Checksum Block cipher

We now introduce some information about CRC. CRCs are specifically designed to protect against the errors on communication channels, and they can ensure the integrity of messages on the channel. However, they are not suitable for protecting against intentional alteration of data. An attacker can change messages and compute the CRC again without detecting in unauthorized situation. CRCs and cryptographic hash functions do not protect against intentional modification of data when stored alongside the data. It must use cryptographic authentication mechanisms to protect against attacks in any application, such as message authentication codes or digital signatures. CRC is a linear function with a property that $\text{crc}(x \oplus y \oplus z) = \text{crc}(x) \oplus \text{crc}(y) \oplus \text{crc}(z)$.

In the homomorphic decryption procedure, the Checksum(Ptxt_before) does the XOR operation with the result of block cipher decryption to get plaintext. The plaintext includes user's input data and the checksum of data, all of them are performed homomorphic encryption. Then, the userNode checks whether the checksum of plaintex is equal to the checksum of plaintext which is performed decryption from ciphertext. If Checksum(Ptxt_before) is equal to Checksum(Ptxt_after), the check of data integrity is successful, otherwise the computation is failed and terminated.

We now describe about the usual definitions for blockciphers and their security. A blockcipher is a function $E : \kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ such that $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\{0,1\}^n$. The inverse to the blockcipher E is $D = E^{-1}$ defined by $D_K(b)$ being the unique $a \in \{0,1\}^n$ such that $E_K(a) = b$.

The way to make explicit explanations for the security of a blockcipher $E : \kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ works as follows. Choose a random $K \xleftarrow{\$} \kappa$ and a random permutation π on n bits. An adversary $\hat{A}$ is given black box access either to E_K or to π. The adversary tries to guess which kind of object it has. Equation 1 is depicted as follows.

$$\text{Adv}_E^{\text{prp}}(\hat{A}) = \Pr[K \xleftarrow{\$} \kappa : \hat{A}^{E_K(\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \text{Perm}(n) : A^{\pi(\cdot)} \Rightarrow 1] \quad (1)$$

Another measure of security for a blockcipher is to compare against a pseudo random function (PRF) instead of a pseudo random permutation (PRP). In our case, Equation 2 is presented as follows.

$$\text{Adv}_E^{\text{prf}}(\hat{A}) = \Pr[K \xleftarrow{\$} \kappa : \hat{A}^{E_K(\cdot)} \Rightarrow 1] - \Pr[\rho \xleftarrow{\$} \text{CRC}(n) : \hat{A}^{\rho(\cdot)} \Rightarrow 1] \quad (2)$$

where CRC(n) denotes the set of all checksum functions from checksum of n bit strings to checksum of n bit strings. It is known that $\Pr[\hat{A}^{\pi} \Rightarrow 1] - \Pr[\hat{A}^{\rho} \Rightarrow 1] \leq q^2/2^{n+1}$ for any adversary $\hat{A}$ that makes at most q queries. This makes the PRP and PRF notions of advantage close: $|\text{Adv}_E^{\text{prp}}(\hat{A}) - \text{Adv}_E^{\text{prf}}(\hat{A})| \leq q^2/2^{n+1}$ if $\hat{A}$ makes q or fewer queries. The PRP and PRF notions have been proven to be the most productive in block cipher security.

The following lines are pseudo code about encryption of BVHE (Block Verifiable Homomorphic Encryption).

1: **Procedure** Block Verifiable Homomorphic Encryption
2: **Input**
3: Ptxt: The plaintexts of input from userNode
4: PK: The Homomorphic public key generated by userNode
5: **Output**
6: Ctxt: The ciphertexts are encrypted from userNode's plaintext
7: **Pseudo Code**
8: **Generate** a Public key and a Secrete key;
9: **While** NOT reach the end of usernode's input Ptxt
10: **While** the block slots is not full **do**

```
11:     add Ptxt and the checksum of Ptxt into block ;
12:     end while
13:       do Encryption procedure(PK, Ptxt_block);
14:   end while
```

The following lines are pseudo code about decryption of BVHE (Block Verifiable Homomorphic Encryption).

```
1:   Procedure  Block Verifiable Homomorphic Decryption
2:   Input
3:    Ctxt: The ciphertexts are encrypted from userNode's plaintext
4:    SK: The Homomorphic secrete key generated by userNode
5:   Output
6:    Ptxt: The plaintexts from userNode
7:   Pseudo Code
8:    While NOT reach the end of usernode's input Ctxt
9:      While the block slots is not full do
10:     add Ctxt into block;
11:     end while
12:       do Decryption procedure(SK, Ctxt_block);
13:     do data integrity procedure(CRC(Ptxt_before), CRC(Ptxt_after));
14:   if integrity is true then return Ptxt
15:   else stop ALL;
16:   end while
17:   do verify computation procedure;
```

3.2.2 Verifiable Computing Scheme

In this scheme, the user must still perform an expensive one-time preprocessing. After that, the client runs in linear time. It is important to emphasize that it can be executed in a trusted situation, so the users who have not enough power to computation can outsource their work to a trusted party.

We now describe how the scheme and the protocol work. We use Yao's garbled circuit to generate key, and the calculation runs over a Boolean circuit computing the function F, it uses the set of garbled version ciphertexts as public key, and it uses all the garbled version random wire labels as secret key. The input is encoded in two steps: first client needs to generate a pair of keys by the homomorphic encryption scheme, and then the labels of each input wire are encrypted with it. The garbled version of the ciphertexts are consist of the public encoding of the input, and the garbled version secrete key is kept private by the user.

3.3 Threat Model

In our verification scheme, we assume it to be semi-honest which means two parties follow the protocol. So there is no need to use cut-and-choose methods for verifying the garbled circuit which is constructed by Alice.

3.3.1 Three phases of verifiable computation

Phase 1 - Preprocessing

The user computes some auxiliary public and private information for function F in this phase.

Phase 2 - Input Preparation

The client prepares some auxiliary public and private information about x, called σ_x and τ_x in this phase.

Phase 3 - Output Computation and Verification

The evaluator uses the public information associated with F and x. It computes a string π_x, the string encodes the value $F(x)$ and returns it to the user. The user can compute the value $F(x)$ and verify its correctness by the value π_x.

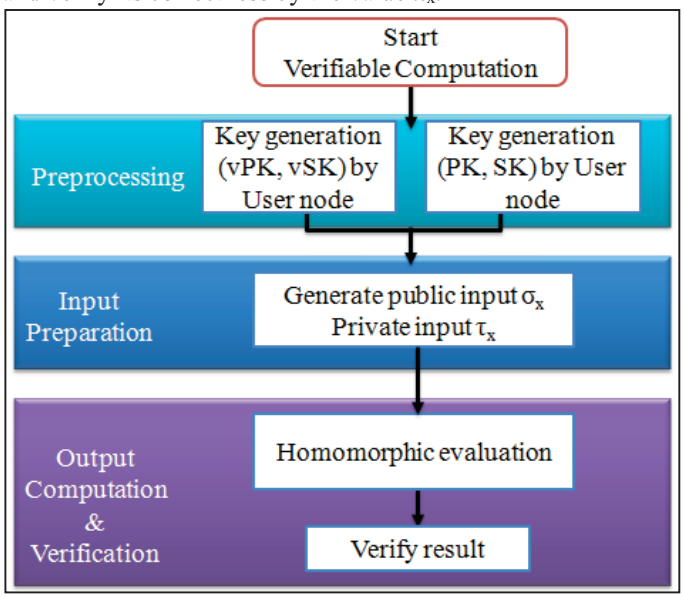


Fig. 6. Verifiable computation scheme flow of Block Verifiable HE.

Figure 6 shows the verifiable computation scheme flow of Block Verifiable HE. This flow involves the following steps:

a) When verifiable computation procedure starts, it does some process in the preprocessing stage. The userNode generates a pair of key for homomorphic computation, generates ciphertexts, $\cup_g({}^{00}\gamma_g, {}^{01}\gamma_g, {}^{10}\gamma_g, {}^{11}\gamma_g)$ for public key vPK of verifiable computation, and uses chosen wire values $\cup_i({}^0w_i, {}^1w_i)$ to be the secrete key vSK of verifiable computation.

b) The userNode uses the secret key SK to encode the function input x as a public value σ_x which is given to the workNode to compute with, and a secret value τ_x which is kept private by the userNode.

c) Using the userNode's public key and the encoded input, the workNodes compute an encoded version of the function's output $y = F(x)$.

d) Using the secret key SK and the secret decoding τ_x, the verification algorithm converts the workNode's encoded output into the output of the function, thus $y = F(x)$. Otherwise, it outputs $\perp$ indicating that σ_y does not represent the valid output of F on x.

The following lines are pseudo code about verifiable computation scheme of BVHE(Block Verifiable Homomorphic Encryption).

1: **Pseudo Code**
2: **do** KeyGen(F,λ) → (PK,SK) ;
3: **For** (i = 1; i<= ℓ=poly(λ); i++)
4: **do** ProbGenSK(x_i) → (σ_i , τ_i) ;
5: **end For**
6: **do** Compute$_{PK}$(σ_x) → σ_y ;
7: **do** Verify$_{SK}$(τ_i ,σ_y) → y;
8: **if** y = ⊥ **then**
9: **return** False;
10: **elseif** y = F(x_i) **then**
11: **return** True;

4. Implementation and Experiment

4.1 Experiment environment and setting

In the experiments, we built a peer-servicing cloud computing platform with 11 homogeneous virtual machines. The hardware and software specifications are detailed in Table I and Table II respectively.

TABLE I
SYSTEM ENVIRONMENT

Item	Content
OS	Ubuntu 12.04LTS Desktop 64bit
Hadoop	1.2.1
Java	1.7.0_76
GCC	gcc version 4.6.3(Ubuntu/Linaro 4.6.3-1ubuntu5)

TABLE II
HARDWARE SPECIFICATIONS

Item	Content
CPU	Intel(R) Xeon(R) E5620 @2.40GHz x 2
RAM	8 GB
Hard Drive	80GB
Network Bandwidth	1Gbps

This study implements enhanced FHE model under two scenarios, secure SMS filtering and secure stock evaluation. And we compare the original method of homomorphic encryption with the proposed mechanism.

4.2 Scenario and Application of experiment

In order to evaluate our proposed scheme, few evaluation scenarios are defined and elaborated in this section. We explain the evaluation scenarios and assumptions in this section.

4.2.1 Secure Stock Evaluation

First, we discuss the scenario of secure stock evaluation using our FHE model and using our block cipher FHE scheme which encrypts and decrypts data block by block as Figure 7 shows. We use three different amounts of records, 1024 records, 2048 records, and 4096 records, each record in this case is 7 bytes. And different colors of line chart present different secure degrees. The blue line chart is for 80-bit key length security, and the orange line chart is for 128-bit key length security.

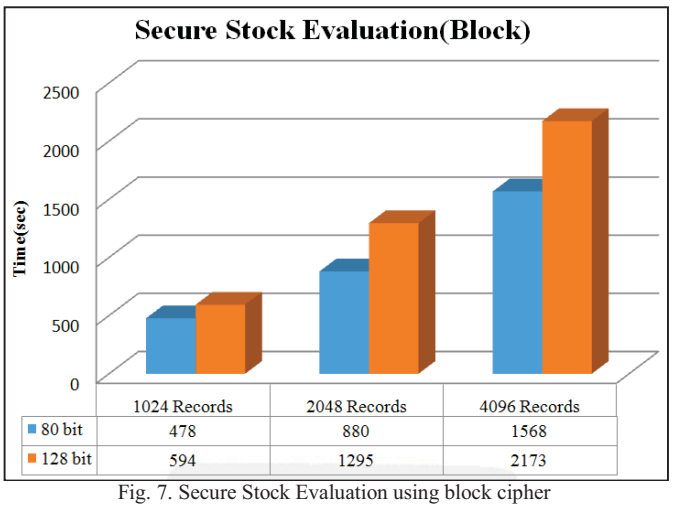


Fig. 7. Secure Stock Evaluation using block cipher

And then we discuss the scenario of secure stock evaluation using our FHE model and using original FHE scheme which encrypts and decrypts data one by one as Figure 8 shows. We use three different amounts of records, 1024 records, 2048 records, and 4096 records, each record in this case is 7 bytes. And different colors of line chart present different secure degree. The blue line chart is for 80-bit key length security, and the orange line chart is for 128-bit key length security. We can observe that the time spent is proportional to the amount of records such that processing double input record needs nearly double cost of time.

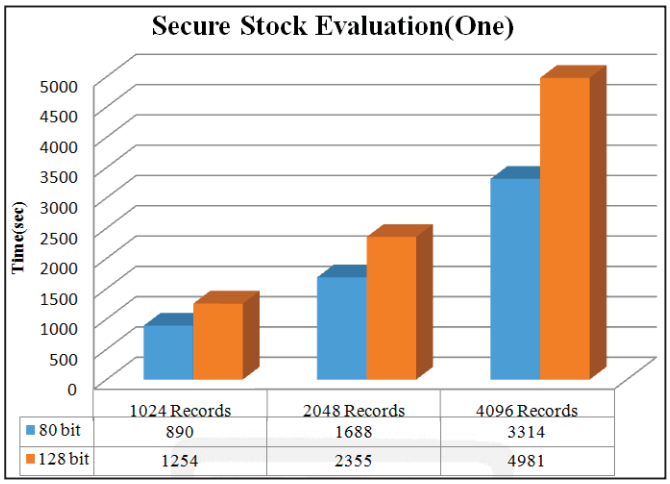


Fig. 8. Secure Stock Evaluation using original cipher

4.2.2 Secure SMS Filtering

The experiment result of SMS filtering using our FHE model and block cipher FHE scheme, which encrypts and decrypts data block by block, are drawn in Figure 9. Three different amounts of records are tested. Each record in this case is 8 bytes. And, different colors of line chart present different secure degrees. The blue line chart is for 80-bit key length security, and the orange line chart is for 128-bit key length security. We observe that the more scale of input records the more time spent. For example, processing quadruple input record needs nearly quadruple cost of time.

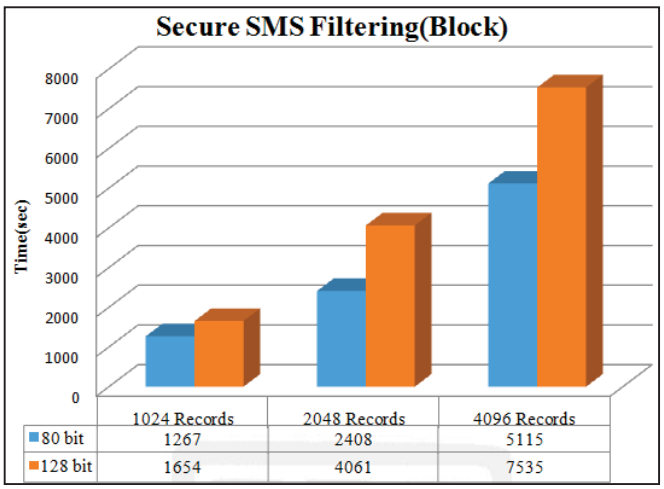


Fig. 9. Secure SMS Filtering using block cipher

And then we discuss the scenario of secure SMS filtering using our FHE model and using original FHE scheme, which encrypts and decrypts data one by one, as Figure 10 shows. We use three different amounts of records, 1024 records, 2048 records, and 4096 records. Each record in this case is 8-byte long. And different colors of line chart present different secure degrees. The blue line chart is for 80-bit key length security, and the orange line chart is for 128-bit key length security. We can observe that the more scale of input records the more time spent. For example, processing double input record needs nearly double cost of time.

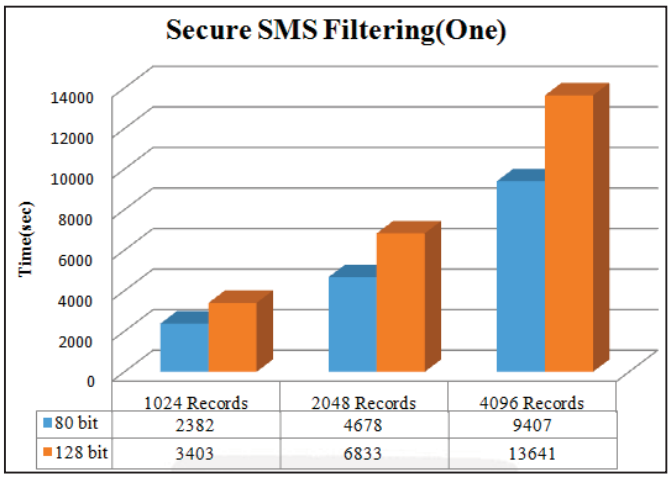


Fig. 10. Secure SMS Filtering using original cipher

Besides, we can also observe that the cost time of our FHE model is less than that of the original FHE model. For example, in the case of input size of 4096 records with 128-bit secure key, the original FHE model spends 13641 seconds, while our FHE model only takes 7535 seconds.

5. Conclusions and Future Work

We apply three scenarios to our FHE verifiable scheme: secure stock evaluation, secure SMS filtering, and secure satisfaction evaluation. AES encrypts information by repeatedly using the following four kinds of data transformation: ShiftRows, SubBytes, MixColumns, and AddRoundKey. We also take two steps, SubBytes and MixColumns, to compare our enhanced cryptosystem with original homomorphic scheme. Our scheme encrypts and decrypts blocks by blocks with checksum for data integrity. On the other hand, the original scheme is performed one integer by one integer (or one char by one char) to perform encryption or decryption. We also observe that the cost time for our FHE model is less than that for the original FHE model.

In the future work, the homomorphic encryption and decryption can be more efficient. Pipeline computing model and the support of GPU could be implemented to improve performance of our system. The method of checking data integrity can be replaced with the one which is more efficient and more robust. Namely, the method must be chosen to maximize the error-detecting capabilities while minimize overall collision probability.

6. References

[1] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM, 51 (1)*, pp.107–113.

[2] Shvachko, K. , Kuang, H. , Radia, S. , and Chansler, R. , "The hadoop distributed file system. In Proceedings of the IEEE Symposium on Massive Storage Systems and Technologies," *IEEE*, Los Alamitos, CA, 2010.

[3] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis, "Evaluating mapreduce for multi-core and multiprocessor systems," in *HPCA'07*, Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture, Washington, DC, USA, 2007, pp. 13–24.

[4] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes". in *18th Annual Eurocrypt Conference (EUROCRYPT'99)*, Prague, Czech Republic, *volume* 1592, 1999.

[5] Julien Bringe & al. (2007) An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, *Springer-Verlag*.

[6] R. Rivest, A. Shamir, & L. Adleman. (1978) A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM,21(2)*,pp. 120-126.

[7] Taher ElGamal. (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, pp. 469-472.

[8] Craig Gentry & Shai Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," Manuscript, to appear in *FOCS*, 2011.

[9] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography," in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, 2005, pp. 84-93.

[10] Chris Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in Proceedings of the 41st annual ACM symposium on Theory of computing, 2009, pp. 333-342.

[11] Peikert, Chris, Mosca, and Michele, ed. "Lattice Cryptography for the Internet," Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 197–219. ISBN 978-3-319-11658-7.

[12] A. Yao, "Protocols for secure computations," in Proceedings of the IEEE Symposium on Foundations of Computer Science, 1982.

[13] Kreuter, B., Shelat, A., Shen, C.H, "Billion-gate secure computation with malicious adversaries," USENIX Association, 2012, pp. 14–14.

[14] Y. Lindell & B. Pinkas. (2009) A proof of Yao's protocol for secure two-party computation. *Journal of Cryptology, 22(2)*, pp. 161–188.

[15] Marozzo F, Talia D, and Trunfio P. P2P-MapReduce: parallel data processing in dynamic cloud environments. *J. Comput. System Sci. 2012;78(5)*, pp. 1382–402.

[16] Stallings and William, "Cryptography and Network Security," 6th ed., Upper Saddle River, N.J.: Prentic Hall, 2014, pp. 67–68. ISBN 978-0133354690.

Applications and Evaluation of Ambiguous Multi-Symmetric Cryptography in Vehicular Ad-hoc Networks

Ahmad Mansour
Department of Computer Science and Engineering
Oakland University
Rochester, Michigan
aamansour@oakland.edu

Kevin Miller
Department of Computer Science and Engineering
Michigan State University
Lansing, Michigan
mill2322@msu.edu

Yakeen Alwishah
Department of Computer Science
Wayne State University
Detroit, Michigan
eq3745@wayne.edu

Richard Bassous
Department of Computer Science and Engineering
Oakland University
Rochester, Michigan
rbassous@oakland.edu

Huirong Fu
Department of Computer Science and Engineering
Oakland University
Rochester, Michigan
fu@oakland.edu

Ye Zhu
Electrical Engineering and Computer Science Department
Cleveland State University
Cleveland, OH USA
y.zhu61@csuohio.edu

Abstract— **Vehicular Ad-hoc Networks (VANETs) are an anticipated mobile ad-hoc system that allows communication between vehicles and roadside infrastructure to increase safety on the road. It is essential that this system is able to send fast and secure messages to prevent accidents. In this research, we apply the Ambiguous Multi-Symmetric Cryptographic (AMSC) primitive in four different models to vehicular ad-hoc networks. The goal of these models is to ensure the confidentiality of messages using a secure encryption scheme.**

Keywords- Symmetric Cryptography; Multi Encryption; Multicast; Chinese Remainder Theorem; VANETs.

I. INTRODUCTION

Vehicular Ad-hoc Network (VANET) builds on features of Mobile Ad-hoc Networks (MANETs) to create a network of vehicles on the road to communicate safety messages and disseminate information. Safety messages are the messages that provide an important and critical information relating to road conditions and driving safety on the road for all vehicles on the network. The purpose of these messages is to make the vehicle aware about the surrounded environment and other vehicle neighbors. VANETs are expected to vastly improve safety for vehicles on the road by reporting vehicle information such as velocity and position. Currently, the only agreed upon standards for VANET in the United States are defined by the IEEE WAVE 1609 protocol stack which implements reserved channels in the 5.9 GHz frequency band defined by IEEE 802.11p WLAN; however, there is still need for a comprehensive protocol that addresses private, secure, and efficient message transfer between vehicles or infrastructure.

Cryptography is used to secure communication by encrypting and decrypting messages. An attacker who intercepts a message will not be able to interpret the message or deduce sensitive information. There are many different cryptographic methods for encryption and decryption. In the modern age, cryptography primarily utilizes mathematics for encryption and decryption.

There are two studies of cryptography that VANETs are primarily concerned with; symmetric key cryptography which uses a shared secret key between the two communicating nodes, and asymmetric cryptography which consists of public and private key pairs for each node. Symmetric cryptography uses a symmetric algorithm to encrypt a message with the shared secret key. When the encrypted message is received, the receiver will decrypt using the algorithm with the shared secret key. Asymmetric cryptography uses an asymmetric algorithm to create a public and private key pair for a node. The node can then broadcast its public key. To send a message to that node, a different node encrypts the message using the asymmetric algorithm and the public key. That message can only be decrypted using the corresponding private key for that node. A caveat of using symmetric cryptography is that the secret key must be shared ahead of time, but by using asymmetric cryptography, all other nodes only need access to the public key for a particular node. For this reason, asymmetric cryptography is more scalable for memory complexity than symmetric cryptography; many nodes can exist on the network without causing massive memory consumption for storing keys unlike symmetric cryptography. Although memory complexity

is worse for symmetric cryptography, encryption and decryption is faster if an appropriate algorithm is selected.

Because VANETs are a highly mobile and dynamic environment, symmetric cryptography can be faster and easier to implement encryption and decryption of safety messages. We explore the use of the Ambiguous Multi-Symmetric Cryptographic (AMSC) primitive [1] in four models. We address some solutions to the concerns associated with memory complexity when using a symmetric algorithm.

Secure communication is a requirement for VANETs, but privacy must also be considered when choosing a protocol for communication. Because VANETs will constantly report position and velocity, it is essential that an attacker cannot track a vehicle without physically following them. There are several protocols that propose non-cryptographic methods for maintaining privacy for individual vehicles so they cannot be tracked. For example, dummy messages, silent periods, mix-zones, and cloaking [2] are all non-cryptographic means of ensuring privacy. Our paper explores using or combining cryptography with vehicle clustering, data aggregation, or non-cryptographic methods for privacy.

The remainder of this paper is organized as follows: Section II briefly explains how the AMSC algorithm works. Next, Section III details the modes of communication and communication paths used in the models we propose. Then, Section IV describes in detail the four models of our protocol. Finally, Section V concludes.

II. AMSC ALGORITHM

AMSC [1] is a cryptographic primitive that encrypts *n* different plaintexts using *n* different keys to produce one ciphertext. Each receiver has *1* or more of the *n* keys, in which it can use to decrypt the ciphertext to get back the original plaintext. Multiple decoy keys are used to deny encryption [1]. AMSC can broadcast one message and be decrypted to different messages at each receiver. Two phases for AMSC are as follows.

A. Encryption

In AMSC, *n* different keys are generated that are co-primes with each other. Every key has to be bigger than its plaintext as well. The encryption uses the Chinese Remainder Theorem to encrypt *n* different plaintexts with their corresponding keys and generate one ciphertext which conceals all plaintexts [1].

B. Decryption

The decryption is simple. Take the ciphertext and mod it with a specific key K_i to get back the original plaintext message M_i. Each receiver could have multiple keys that each decrypts the same cipher to a different coherent message.

C. Message Dissemination

The AMSC algorithm allows for messages to be disseminated in different ways, depending on what needs to be accomplished. First, AMSC can send *n* real messages encrypted to many receivers. These messages will be encrypted as a single ciphertext. Each receiver can decrypt their own message with their designated shared secret key. See Figure 1 for a one-to-many broadcast (multicast) of AMSC.

The AMSC primitive can also be used to send decoy messages. If there is one real message that needs to be sent, then we can encrypt the ciphertext with one real message and *n-1* decoy messages. In the event that an attacker eavesdrops to retrieve and decrypt the ciphertext, they will not be able to know if the decrypted message is the correct real message. For example, if the real message said "Meet at the library", we can encrypt several equally plausible decoy messages like "Meet at the gas station" and "Meet at the office". This ensures that the attacker will not be able to tell the decoy messages from the real one. See Figure 1 for a multicast AMSC with one real message and *n-1* decoy messages.

AMSC can also send one message to only one receiver. First, the sending node and receiving node must share *n* keys where $n > 1$. A message *M* that needs to be sent to a single receiver can be divided into *n* smaller parts so that each part of the original message or plaintext becomes *M/n*. Each part will be encrypted with a separate shared key in parallel to become one ciphertext. Afterwards, the ciphertext can be decrypted using the corresponding keys on the receiver side to retrieve *n* plaintext messages $M_1..M_n$. Finally, these plaintexts are concatenated to get the original message *M*. Using AMSC in parallel can significantly speed up the encryption and decryption process since the message size is made smaller and each message is encrypted in parallel. See Figure 2 for one-to-one communication of AMSC in parallel.

D. AMSC for VANETs

VANET systems will have particular requirements for messages including position, velocity, destination coordinates, vehicle ID, and more. While the easiest option is to simply send a message without encryption, it is prudent to protect messages from tampering, eavesdropping, interception and other attacks by using a cryptographic algorithm; however, adding this level of security comes at the cost of memory to store messages and keys, and speed of communication. Because of this trade-off, it is important to find a cryptographic algorithm that balances security, speed, and memory so they are maximized.

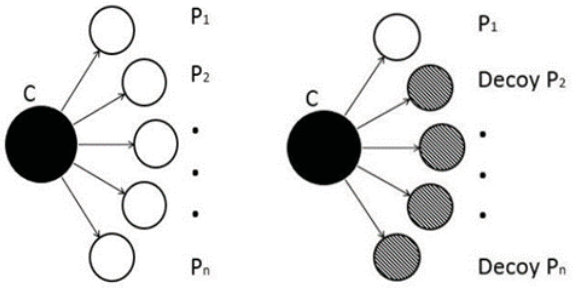


Figure 1. Multicast with n Real Messages (left), Multicast with *1* Real Messages and *n-1* Decoy Messages (right).

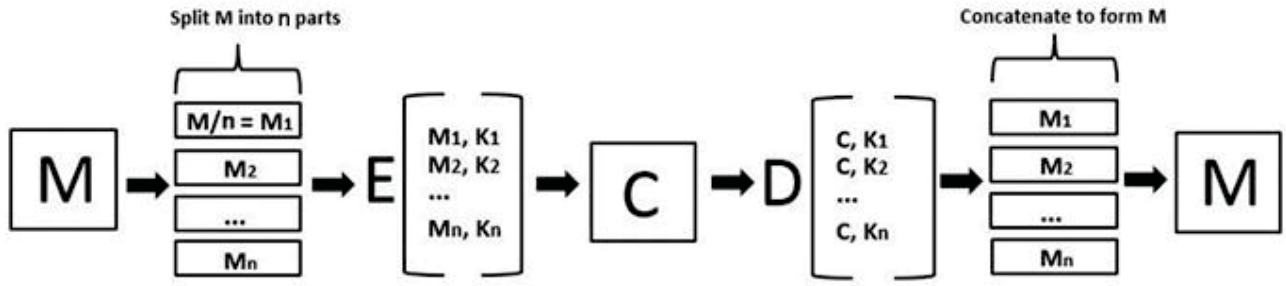


Figure 2. One-to-one Communication of AMSC in Parallel.

The AMSC algorithm is an ideal choice because of its superiority in speed and security when compared to other symmetric algorithms. Furthermore, AMSC provides ambiguity which enhances security further. AMSC also provides a way to exchange data from one source to many receivers which is valuable in a VANET system where one vehicle or roadside unit may want to communicate to a number of other vehicles. This feature becomes very useful in some of the models we propose later in this paper as the AMSC algorithm is able to send one real message and *n-1* decoy messages. In this way, we can use AMSC in an event where dummy messages need to be sent out. Because of the advantages and added utility of using AMSC, we chose to use it instead of other cryptographic algorithms.

III. COMMUNICATION

This section will detail the modes of communication that AMSC handles in VANETs and the possible paths of communication. Clearly defining these characteristics allows our proposed models to be compared and an appropriate model for the VANET setting to be determined. Since VANETs are expected to allow communication between vehicles and roadside infrastructure, we analyze how vehicles and infrastructure could interact with each other using AMSC for encryption and decryption. Also, we assume that the secret keys have been distributed using one of the standard key distribution schemes.

A. Modes of Communication

AMSC can broadcast messages in three different ways. The first one is a basic multicast where one source will broadcast messages to many receives (one-to-many). The second type of broadcasting in AMSC is also a multicast, but it has only one real message and *n-1* fake messages. Using this method, a potential attacker will not be able to discern between a real message and a fake message. The final way AMSC can broadcast messages is from one source to one receiver.

B. Communication Paths in VANETs

Communication in VANETs can go in one of three ways: infrastructure-to-vehicle(s) (I2V), vehicle(s)-to-infrastructure (V2I), and vehicle(s)-to-vehicle(s) (V2V).

1) Infrastructure-to-vehicle(s): Infrastructure will consist of deployed Roadside Units (RSUs) to assist the VANET system by communicating with cars and connecting them to a larger network. Infrastructure is costly and will be difficult to implement throughout the VANET system, particularly in the early stages of deployment [3]. Because of this, we consider models that do not rely on infrastructure to be present everywhere. VANET can trust the RSUs more than individual vehicles because they are implemented by the government. Vehicles refers to the individuals that will be driving with a VANET system installed. In this communication path, RSUs will encrypt a message for individual vehicles or clusters of vehicles.

2) Vehicle(s)-to-infrastructure: Individual vehicles or clusters of vehicles will encrypt messages for RSUs. In some cases, the infrastructure will proceed to send the information to other vehicles.

3) Vehicle(s)-to-vehicle(s): Individual vehicles or clusters of vehicles will encrypt messages and communicate them directly to other nearby vehicles or clusters. This is essential for time dependent safety messages that could prevent an accident such as hard braking of a vehicle in front of another. Direct transmission is faster than communication through an RSU since the message is only encrypted and decrypted once. Vehicle to Vehicle can also form an ad-hoc network if infrastructure is not present, allowing the VANET system to work even if infrastructure is not constructed everywhere.

IV. AMSC IN VANET PROTOCOL

The AMSC protocol for VANET explores four possible models that will be used to encrypt and decrypt messages, and handle the flow of information in the system. We examine the use of aggregation and clustering for message communication, and determine the shared keys, associated characteristics, advantages, and limitations of these models [9] [10]. Note that whenever two nodes share a key, they could share more than one separate key for one-to-one communication.

A. Model 1 - Clustering Vehicles with a Cluster Head for Aggregation

Vehicles driving on the road will form clusters or groups of vehicles using a clustering mechanism. Some of these protocols form clusters statically by using preloaded map dissections such as in [4], while others suggest dynamic group formation by comparing metrics like velocity, destination, and

current position of nearby vehicles as seen in [5]. If cluster aggregation is desired, a protocol must be agreed upon. [6] compares many different models for clustering which should be considered when deciding on an appropriate protocol for the VANET environment. All members of a cluster will share separate secret keys with each other. During cluster formation, a node will be a distinguished cluster member known as a cluster head. The cluster head will share keys with all of its cluster members as well as neighboring cluster heads and any present RSUs. A diagram of all possible shared keys in model 1 can be seen in Figure 3.

The cluster head is responsible for communication to and from the cluster with vehicles and infrastructure outside the cluster. All vehicles will send messages to the cluster head who will then aggregate and encrypt the message, deleting duplicate messages in the process. While duplicate messages will be deleted, each vehicle that reports the same message will have their signature attached to the message so that others can see how many vehicles agree with the message. This can help verify that a particular message actually occurred. The aggregated and encrypted message can be sent to any nearby clusters or an RSU. For messages being sent to other clusters, it is the responsibility of the receiving cluster heads to decrypt the incoming message and send it to each vehicle in its cluster encrypted with their shared key. Because the cluster head is responsible for communication beyond the group, it is essential that the cluster head can be trusted. There are existing proposals for trust systems and defense from intrusion attacks, such as [7] and [8]; however, it is outside of the scope of our research to determine a trust system. Finally, since each cluster member shares a unique secret key, vehicles can quickly and privately communicate to each other without contacting a third party. This is critical for time sensitive safety messages.

1) Characteristics: Model 1 uses multicast with *n* real messages and multicast with a number of decoy messages is used from RSUs to clusters (I2V), cluster to clusters (V2V), and cluster member to other cluster members (V2V). One-to-one communication is used between two vehicles within a cluster (V2V), between two cluster heads (V2V), or between the RSU and a cluster head (V2I or I2V).

2) Advantages: Clusters provide location privacy for the members of the group since outside attackers cannot determine who is in a cluster or how many vehicles are in the cluster. All that can be determined is that the cluster exists.

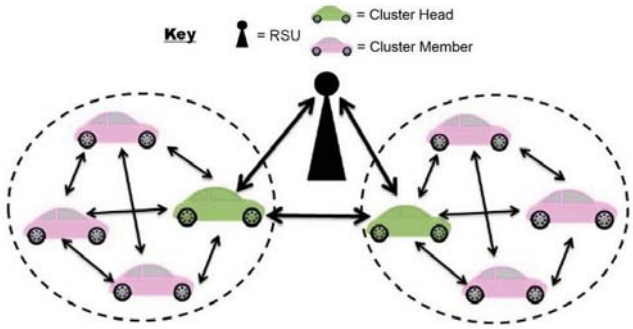


Figure 3. Shared Keys for Model 1.

As stated above, scalability is an issue when considering symmetric algorithms since unique keys must be shared between all nodes in a network. Without a clustering model, if a vehicle wanted to communicate to nearby vehicles directly, it would need to share a key with every vehicle in its range. Furthermore, RSUs would need to share keys with every vehicle in its range as well. In model 1, we limit the amount of shared keys in the system, making AMSC viable. Vehicles only need to share keys with other vehicles in their cluster, cluster heads will need to share additional keys with other cluster heads and the RSU, and RSUs only share keys with cluster heads. Symmetric cryptography is desired in VANETs because of the speed advantages it provides, but it comes at the cost of storing many keys. By solving the scalability issue, we have made AMSC feasible and extremely useful for communication.

Safety messages are a primary concern for VANETs. Since vehicles can directly communicate with each other, these time sensitive messages can be sent quickly, preventing accidents. Clusters can also report an upcoming traffic hazard such as a traffic jam or mudslide to clusters further back so that vehicles can be prepared to exercise caution.

Another advantage of this model is that it does not rely on infrastructure such as RSUs to work [3]. Notes that RSUs are costly and may not be deployed in all locations necessary. Therefore, it is prudent to consider a system that can operate without the existence of RSUs.

Finally, this model deletes duplicate messages in the aggregation process that would consume unnecessary bandwidth. For example, consider an event where an accident occurs. All nearby vehicles would report the same message. Instead of sending all of the messages, attach their signatures to the same message and send.

3) Limitations: A shortcoming of this model is that because the cluster head is responsible for communication with other clusters and the RSU, it is essential for the cluster head to be trusted. Therefore, there is a need for trust protocol, such as [8], to prevent attacks that may start from a malicious cluster head.

Another limitation of this model is that a clustering protocol must also be agreed upon. Different protocols will prove useful depending on the environment and since VANETs are highly dynamic, it may be difficult to arrive at a single clustering protocol. For example, a clustering protocol for dense traffic in a city following a baseball game may differ from a protocol on a rural country road. Clustering may also increase the complexity of the VANET protocol.

Although this model is scalable as a whole and there are advantages to sharing keys with neighbors, a possibility is to strictly share keys with an RSU. This means each car would only have to share one key or set of keys with the RSU which consumes less memory for vehicles than sharing keys with neighboring vehicles within a cluster as proposed in model 1.

B. Model 2 – Clustering Vehicles with a Shared Key and No Cluster Head

In model two of our protocol, vehicles driving along the road will either form clusters through preloaded map dissections or dynamically by comparing velocity, position, and destination or other metrics. During the cluster formation, a group key is distributed to all members and shared with the RSU. The RSU also shares keys with every other cluster in its range, as well as every individual vehicle. When the RSU wants to send a message to every car in a cluster, it can simply encrypt with that clusters key and every car in that cluster can decrypt with their key. Vehicles could also share unique secret keys with each vehicle in its cluster for private, direct communication. A model of these shared keys can be seen in Figure 4. Note that a solid circle surrounding the vehicles represents a single group key that all vehicles in the cluster and the RSU share.

1) Characteristics: The RSU can use multicast to communicate to all of the clusters in its range (I2V), or multicast to communicate with all vehicles in its range (I2V). The RSU can use multicast with a number of decoy messages as well for communication with a vehicle or a cluster. The RSU can use one to one to communicate with a single cluster (I2V) or a particular vehicle (I2V) within a cluster. The vehicles can use one to one communication to communicate with individual cars if they share secret keys (V2V). One to one communication can also be used with the group key to communicate from one vehicle to the whole group (V2V). Nearby clusters could share secret keys with each other, although this may introduce security concerns since any vehicle can communicate on behalf of the group.

2) Advantages: Since this model uses clustering, there is location privacy for individual vehicles since outside observers can merely see that a cluster exists, not who is sending the message unless the RSU is compromised. Additionally, since the RSU is aware of every vehicle in its range, instead of only the clusters, it can keep track of individual vehicles and introduce a justice system if a vehicle acts maliciously.

Safety messages can be sent quickly to individual vehicles if all vehicles within a cluster share unique keys. Furthermore, a cluster member can alert the entire group of a safety concern very fast using the group key in parallel. This is useful if a vehicle detects a concern that affects the entire area such as a fallen tree. These messages can be sent quickly to other nearby clusters if they share a key, although caution should be exercised because any vehicle can speak on behalf of the group.

This model does not rely on a trusted cluster head so the group cannot be compromised by a single malicious vehicle, and there is no need to determine a trust model. This means that an RSU will have to supervise all vehicles since any malicious vehicle could compromise the cluster.

3) Limitations: This model does not necessarily rely on an RSU if vehicles share keys with nearby clusters, however this introduces security vulnerabilities by allowing any car to communicate with other clusters on behalf of the group. Vehicles could share secret keys with every vehicle in its range, including vehicles not in its cluster; however, this will increase memory usage for storing keys for the system. Therefore the preferable method is to rely on the existence of RSUs which may not always be present.

This model does not aggregate messages from a cluster since there is no cluster head, therefore this model does not delete duplicate messages, thus increasing bandwidth consumption.

If the RSU wants to send a multicast to many vehicles in separate clusters, the resulting ciphertext from using one-to-one AMSC in parallel may be too large to use. This contrasts model 1 where the ciphertext size will be significantly smaller since the RSU only sends to each cluster, not individual vehicles.

Lastly, scalability is a concern of this model. Vehicles could share keys with all vehicles in its range, including other clusters and the RSU, but this will mean that the system will have to account for many shared keys. The RSU will also need to be aware of all individual cars which will require sharing many keys.

C. Model 3 – Clustering Vehicles with No Shared Key and No Cluster Head

Vehicles will form clusters through preloaded map dissections or dynamically using a clustering protocol; however, there is no cluster head or shared group key in this model. Vehicles will find it preferable to share keys only with vehicles in their cluster and with the RSU, instead of all vehicles in their range. The RSU will share keys with every vehicle in its range, but will use the clusters to encrypt a ciphertext for multicast. By doing so, the ciphertext will be smaller than sending to all vehicles in its range, which may be infeasible especially in dense traffic. The shared keys are shown in Figure 5.

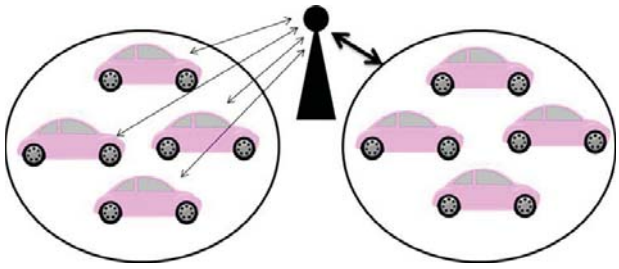


Figure 4. Shared Keys for Model 2.

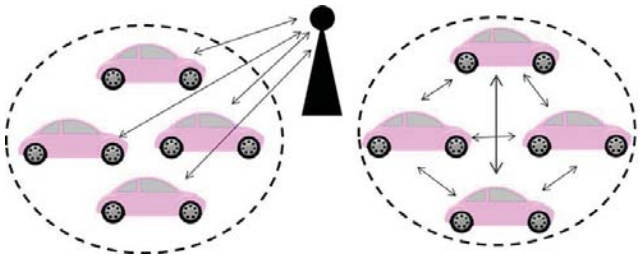


Figure 5. Shared Keys for Model 3.

1) Characteristics: Model 3 can use a multicast when the RSU needs to communicate with the cluster (I2V). The RSU will share unique keys with every vehicle in the cluster. While it is possible for the RSU to communicate with multiple clusters at a time this is not practical since the ciphertext would become very large.

Model three also makes use of the *n-1* fake messages feature of AMSC. If the RSU needs to communicate with a specific vehicle (vehicle A) in a cluster (I2V), it can send a ciphertext to every vehicle in that cluster, but the ciphertext will contain one real message for the intended vehicle and *n-1* decoy messages.

Since the RSU shares a secret key with every vehicle in a cluster in this model, this scheme is also capable of handling communication between the RSU and a particular vehicle and vice versa using one-to-one communication (I2V and V2I). Vehicles can communicate using shared keys with other vehicles, preferably only vehicles within their cluster using one-to-one communication or multicast (V2V).

2) Advantages: This model can send a smaller ciphertext to a particular cluster instead of every vehicle in its range in a model without clustering which is impractical since the ciphertext could be too large. The RSU can also directly communicate with vehicles in a cluster. Additionally, since the RSU is aware of every vehicle in its range, instead of only the clusters, it can keep track of individual vehicles and introduce a justice system if a vehicle acts maliciously.

Vehicles can also send quick secret messages to vehicles in its cluster without introducing any scalability issues associated with sharing keys with all vehicles in its range. This is critical for time dependent safety messages.

3) Limitations: This model relies on RSUs to be effective. If RSUs are not deployed densely, vehicles in this model cannot be tracked and long distance ad-hoc communication between vehicles may be too time consuming or difficult since the vehicle range may not be as large as an RSU range. This scheme also consumes more bandwidth because it does not get rid of duplicate messages since it lacks a cluster head.
Scalability is an issue with this model because the RSU has to share secret keys with every vehicle in a cluster. Furthermore, vehicles are limited to sharing keys within their cluster primarily since sharing keys with other vehicles in its range will require too many key shares. 1.

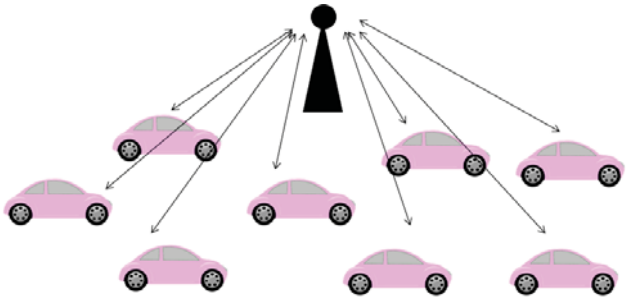


Figure 6. Shared Keys for Model 4.

D. Model 4 - No clustering

This model is simpler in that vehicles do not form clusters. When a vehicle comes in range with an RSU, they share keys. Vehicles could share keys with all vehicles in their range; however, this is impractical when vehicles are dense so it is preferable to only share keys with the RSU. Note that since there is no clustering mechanism, there is no method of determining specific vehicles to share with. The RSU will share keys with all vehicles in its range. These shared keys are shown in Figure 6.

1) Characteristics: The fourth model requires RSUs to share secret keys with all vehicles since there is no clustering involved. An RSU will use one-to-one communication to send an encrypted message to a designated vehicle with a shared secret key and vehicles can communicate with the RSU in the same way (I2V and V2I). Although vehicle-to-vehicle communication is possible, it is not feasible and therefore is not considered. Multicast for the RSU is another possibility, but the ciphertext may become too large especially in dense traffic, so it is not considered either for this model.

2) Advantages: This proposal is one of the simplest ways AMSC can be used since there is no need for a clustering protocol. Furthermore, there is no trusted cluster head, and the RSU can track all vehicles it communicates with. Finally, scalability for vehicles is better than previous models since they only share keys with the RSU and no other vehicles.

3) Limitations: This model cannot operate without RSUs being present since vehicles do not share keys. This is not desired if it is too costly to deploy RSUs at every location they would be needed. Without a cluster head, duplicate messages will not be deleted consuming more bandwidth.

Safety messages will take longer to reach nearby vehicles since they will first be sent to RSU, then to the vehicle. This delay could be detrimental to preventing accidents since the message will be encrypted and decrypted twice before reaching its destination, which could take too long to prevent the accident.

Scalability for the RSU is a concern since it will need to share keys with every vehicle on the road. Furthermore, RSUs cannot communicate with large groups of vehicles since it relies exclusively on one-to-one communication and not multicast.

E. Model Comparison

Each model has unique characteristics, advantages, and disadvantages that make them useful in different circumstances. For example, a highly dense traffic situation following a basketball game in a city may call for a model that implements clustering with the use of RSUs that are put in urban areas to handle the frequent high traffic situations, such as models two or three. A residential area with a lot of traffic may not have RSUs available, but need clustering to manage traffic so model one is preferred. In a remote rural area, model four may be easier to implement because of its simplicity and lack of traffic. We summarize the advantages and

disadvantages for each model to help determine where and when a model might be desired in tables 1, 2, 3, and 4.

Table 1. MODEL 1 - ADVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Location privacy for cluster members.	Relies on a trusted cluster head
Safety messages sent immediately to nearby vehicles in danger.	Clustering increases complexity of VANET architecture.
Does not rely on roadside infrastructure.	Scalability for individual vehicles can be worse than other models.
Decreases unnecessary bandwidth usage by deleting duplicate messages.	
Better scalability for entire system.	
All modes of communication can be used.	

Table 2. MODEL 2 - ADVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Safety messages sent immediately to nearby vehicles in danger.	Relies on roadside infrastructure to send trusted messages.
Does not rely on trusted cluster head.	Clustering increases complexity of VANET architecture.
All modes of communication can be used.	Unnecessary consumption of bandwidth for duplicate messages.
	Untrusted cluster member can speak to other clusters on behalf of the group.
	Poor scalability for roadside infrastructure.

Table 3. MODEL 3 - ADVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Safety messages sent immediately to nearby vehicles in danger.	Relies on roadside infrastructure.
Does not rely on trusted cluster head.	Clustering increases complexity of VANET architecture.
All modes of communication can be used.	Unnecessary consumption of bandwidth for duplicate messages.
	Poor scalability for roadside infrastructure.

Table 4. MODEL 4 - ADVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Simple to implement.	Relies on roadside infrastructure.
Does not rely on trusted cluster head.	Unnecessary consumption of bandwidth for duplicate messages.
Best scalability for individual vehicles out of the four models.	A vehicle cannot privately send a message to another vehicle.
	Safety messages cannot be sent quickly.
	Poor scalability for roadside infrastructure.
	Can only use one-to-one communication.

V. CONCLUSION

In this paper, we proposed a comprehensive protocol that can be used for the efficient distribution of safety messages in Vehicular Ad-hoc Networks. This protocol uses Ambiguous Multi-Symmetric Cryptographic (AMSC) scheme. Our protocol consists of four different models, three of which rely on a clustering protocol, where in the fourth model vehicles communicate directly with the RSUs. We detail each model in its own respective section, making sure to include the advantages and limitations of each model.

ACKNOWLEDGMENT

This research work was partially supported by the National Science Foundation under Grants CNS-1460897, CNS-1338105, and CNS-1343141. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] R. Bassous, R. Bassous, H. Fu, Y. Zhu. "Ambiguous Multi-Symmetric Cryptography". In *Communications (ICC)*, 2015 IEEE International Conference, 2015, pp. 7394-7399.

[2] W. Chen, editor. "Vehicular communications and networks: Architectures, protocols, operation and deployment". *Elsevier*, 2015.

[3] S. Dietzel, J. Petit, F. Kargl, B. Scheuermann. "In-network aggregation for vehicular ad hoc networks". *Communications Surveys & Tutorials*, IEEE, 2014, pp. 1909-1932.

[4] M. Raya, A. Aziz, J. Hubaux. "Efficient secure aggregation in VANETs". In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, 2006, pp. 67-75.

[5] M. Kakkasageri, S. Manvi. "Multiagent driven dynamic clustering of vehicles in VANETs". *Journal of Network and Computer Applications*, 2012, pp. 1771-1780.

[6] R. Bali, N. Kumar, J. Rodrigues. "Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions". *Vehicular communications*, 2014, pp. .

[7] T. Gazdar, A. Benslimane, A. Belghith, A. Rachedi. "A secure cluster‐based architecture for certificates management in vehicular networks". *Security and Communication Networks*, 2014, pp. 665-683.

[8] H. Sedjelmaci, S. Senouci. "An accurate and efficient collaborative intrusion detection framework to secure vehicular networks". *Computers & Electrical Engineering*, 2015, pp. 33-47.

[9] R. Bali, N. Kumar. "Secure clustering for efficient data dissemination in vehicular cyber–physical systems". *Future Generation Computer Systems*, 2016, 476-492.

[10] K. Sampigethaya, M. Li, L. Huang, R. Poovendran. "AMOEBA: Robust location privacy scheme for VANET". *Selected Areas in communications*, IEEE Journal, 2007, 1569-1589.

Cryptanalyses on "secure and efficient privacy-preserving public auditing scheme for cloud storage"

Yalin Chen[1] and Jue-Sam Chou*[2] and Zhe-Yu, Lin[3]
[1] Institute of information systems and applications, National Tsing Hua University

yalin78900@gmail.com

[2] Department of Information Management, Nanhua University, Taiwan
*: corresponding author: jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

[3] Department of Information Management, Nanhua University, Taiwan

10369017@nhu.edu.tw

Abstract

Recently, Worku et al. pointed out that Wang et al.'s "privacy-preserving public auditing for data storage security in cloud computing" is insecure and their second work "privacy- preserving public auditing for secure cloud the storage" is inefficient. Thus, they offered a secure and efficient privacy public auditing scheme for cloud storage. They claimed that their system is provably secure in the random oracle model and the computations are effective. However, after cryptoanalysis, we found that the scheme cannot reach the security goal, it has the existential forgery attack. We, therefore, modify it to include the desired privacy preserving requirement, which is very significant in a privacy-preserving public auditing protocol for cloud storage.

Keywords: *cloud storage, privacy-preserving public auditing, bilinear pairing, signature, file block tag*

1. Introduction

By NIST's definition, cloud computing has five essential characteristics, three cloud service models, and four cloud deployment models. Besides, cloud security alliance (CSA) has identified multi-tenants as an important element of cloud [1]. From the related statements in the literature, we can see that cloud computing environments provide human beings many conveniences, whereas they also bring many problems such as, cloud storage security, due to its multi-tenancy nature and the cloud server may itself be untrustable. In the privacy-preserving public auditing scheme literature, the users don't possess the outsourced data physically. Hence, checking the integrity of the outsourced encrypted data on the cloud server becomes important. There have been many cryptographic works within this field named privacy-preserving public auditing (PPPA) for cloud storage system designs [2-22]. In 2014, Worku et al. [2] pointed out that Wang et al.s' work "privacy-preserving public auditing for data storage security in cloud computing" [3] is insecure and their second work "privacy- preserving public auditing for secure cloud the storage" [4] is inefficient. Therefore, they proposed a secure and efficient one for cloud storage auditing. They claimed that their scheme is provably secure in the random oracle model and the performance is

efficient. However, after cryptoanalysis, we found that the scheme cannot reach the claimed security goal. It has the existential forgery attack. We, therefore, modify it to comprise the desired requirement which is very important in a privacy-preserving public auditing protocol for cloud storage. We demonstrate it in this article.

2. Review of Worku et al.'s auditing scheme

Worku et al.'s public auditing for cloud storage design [2] adopts the framework of an independent third-party auditor (TPA) to audit the outsourced data when needed as does in [3, 4]. It consists of four basic algorithms; KeyGen, SigGen, ProofGen and VerifyProof. In the following, we first briefly describe them in section 2.1. Then, point out the weaknesses in section 2.2. After that, we propose a modification to achieve the desired requirement in section 2.3. As for the used notations, please refer to the original article.

2.1 The four algorithms

(a) KeyGen

The client generates a random signing key pair *(ssk, spk)*, chooses $x \in_R Z_p$, $u \in_R G$, and computes $v = g^x \in G$. He then uses *sk =(x, ssk)* as his secret key and *pk =(u, v, g, spk)* as public parameters.

(b) SigGen

The client chooses a random element in Z_p as the file name $F = \{m_i\}_{1 \le i \le n}$ and computes the file tag t as name||Sigssk(name), where Sigssk(name) is the signature on name. Subsequently, for each block $m_i \in Z_p$ $(1 \le i \le n)$, the user generates a signature $\sigma_i = (H(i) \cdot u^{m_i})^x \in G$ and sends it to the server for storage. Afterwards, the user deletes the file and its corresponding signatures from local storage. Later, when *TPA* wants to start the auditing protocol, it retrieves the file tag *t* and checks its validity using *spk*. If the proof of t is correct, the client or *TPA* constructs and sends a challenge *chal* to the server. That is, *TPA* picks random elements c, k_1, k_2, all in $\in Z_p$, and sends $chal = (c, k_1, k_2)$ to the server, where k_1, k_2 are pseudorandom permutation keys chosen by the user for each auditing.

(c) ProofGen

After receiving *chal*, the server determines the subset $I = \{s_j\}$ $(1 \le j \le c)$ of set $\{1,2,...,n\}$ using pseudorandom permutation $\pi_{key}(\cdot)$ as $S_j = \pi_{k_1}(j)$, and also determines $v_{s_j} = f_{k_2}(j)$ using pseudorandom function $f_{key}(.)$. Finally, for $i \in I$, the server computes:

$\mu^* = \sum_{i=s_1}^{s_c} v_i m_i$,

$\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i}$

Moreover, the server chooses a random $r \in_R Z_p$ for blinding by using the same function *f* and computes $r = f_{k3}(chal)$, where k_3 is a pseudorandom function key generated by the server for each auditing. It then calculates $R = u^r \in G$, $\mu = \mu^* + rh(R) \in Z_p$, and sends (μ, σ, R) to *TPA*.

(d) VerifyProof(*pk, chal*)

Upon receiving the proof (μ, σ, R) from the server, *TPA* computes $S_j = \pi_{k_1}(j)$, $v_{s_j} = f_{k_2}(j)(1 \le j \le c)$, and verifies the proof by checking whether Eq. (1) holds or not, as shown below.

$e(\sigma, g)? = e(\prod_{i \in I} H(i)^{v_i} \cdot u^{\mu} \cdot R^{-h(R)}, v)$ ……Eq. (1)

The correctness can be proved as follows:

$$e(\sigma, g) = \quad e(\prod_{i=s_1}^{s_c} \sigma_i^{v_i}, g) = e(\prod_{i=s_1}^{s_c} (H(i) \cdot u^{m_i})^{x \cdot v_i}, g)$$

$$= e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{v_i m_i}), g)^x = e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\sum_{i=s_1}^{s_c} v_i m_i}, g^x)$$

$$= e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\mu^*}, v) = e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\mu - rh(R)}, v)$$

$$= e(\prod_{i \in I} H(i)^{v_i} \cdot u^{\mu} \cdot R^{-h(R)}, v)$$

If Eq. (1) holds, the proof (μ, σ, R) is valid.

2.2 The weaknesses

As mentioned earlier, the server chooses a random element $r \in_R Z_p$ by using the same pseudorandom function for blinding and let $r = f_{k3}(chal)$, where k_3 is a pseudorandom function

key generated by the server for each auditing. It then calculates $R = u^{r} \in G$, $\mu^{*} = \sum_{i=s_1}^{s_c} v_i m_i$, $\mu = \mu^{*} + \mathrm{rh(R)} \in Z_p$, and $\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i}$.

From the received (μ, σ, R), we can see that since $\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i} = \prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\sum_{i=s_1}^{s_c} v_i m_i})^{x}$, a malicious server can regard $v_i s$ as constants and $m_i s$ as variables. He then computes $\mu^{*} = \sum_{i=s_1}^{s_c} v_i m_i$ using the constants $v_i s$ and the message blocks stored. That is, he can obtain an equation containing multiple variables, the $m_i s$, which in mathematics has more than one solution. This means that other than the original $m_i s$, the malicious server can find out some message blocks satisfying the equation without alerting σ. We take S_c=3 as an example. Suppose the values of $v_i s$ are (6, 8, 9), and the values of $m_i s$ are (1, 4, 2) respectively, then the plane can be defined by $6x + 8y + 9z = 56 (= 6m_1^{*} + 8m_2^{*} + 9m_3^{*})$, where m_i^{*}, *i=1 to 3*, are the forged message blocks. We know that this plane also passes through the point (5, 1, 2). This implies that the malicious server can forge the message blocks from (1, 4, 2) to (5, 1, 2) without alerting the value σ. Moreover due to the independence between $\mu^{*} (= \sum_{i=s_1}^{s_c} v_i m_i)$ and R, any attacker can even set $R' = u^{r'}$ and $\mu' = \mu^{*} + r' h(R') \in Z_p$, and sends (μ', σ, R') to *TPA* after intercepting (μ, σ, R). *TPA* will accept the verification without detection. That is, the proof of the selected blocks is not unique. This might lead the scheme incur more vulnerabilities.

Recently, several articles proposed also have the same problem. For the intested readers, please refer to [18,19,20,21,22].

3. The proposed scheme

From the weaknesses found in section 2, we see that the key point is that the malicious server owns the message blocks and the values of $v_i s$. This is because he can easily find forged message blocks m_i^{*} to satisfy the value $\mu^{*} (= \sum_{i=s_1}^{s_c} v_i m_i^{*})$ without alerting σ. Therefore, we must try to break down the

linear structure of $\mu^*(=\sum_{i=s_1}^{s_c} v_i m_i)$. As a result, we set $\mu^*(=\sum_{i=s_1}^{s_c} v_i m_i h(H(m_i \oplus i)))$ and add one relationship into μ^* and R by setting $\mu = \mu^* + r(h(R) + \mu^*) \in Z_p$ to prevent the found problems.

Certainly, we must first let the client's signature σ_i on m_i to be $(H(i) \cdot u^{m_i h(H(m_i \oplus i))})^{x \cdot}$.

Accordingly, if a malicious server launches the above attack on our modification; although, he knows the values of $v_i s$ and $m_i s$, he cannot break the modification. Thus, the privacy is preserved. The correctness of the verification equation can be shown as follows:

$$e(\sigma, g) = e(\prod_{i=s_1}^{s_c} \sigma_i^{v_i}, g) = e(\prod_{i=s_1}^{s_c} (H(i) \cdot u^{m_i h(H(m_i \oplus i))})^{x \cdot v_i}, g)$$

$$= e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{v_i m_i h(H(m_i \oplus i))}), g)^x = e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\sum_{i=s_1}^{s_c} v_i m_i h(H(m_i \oplus i))}, g^x)$$

$$= e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\mu^*}, v) = e(\prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\mu - r(h(R) + \mu^*)}, v)$$

$$= e(\prod_{i \in I} H(i)^{v_i} \cdot u^{\mu} \cdot R^{-(h(R) + \mu^*)}, v)$$

3 Conclusion

In this paper, we showed that Worku et al.'s work privacy-preserving public auditing for data storage security in cloud computing is flawed. It suffers from the existential forgery attack. Several recent proposed articles also have the same problems which we have mentioned in section 2. For enhancing the security, we therefore modified the scheme to avoid the weaknesses. From the analysis shown in section 4, we see that we have reached the security promotion goal.

References

[1] CSA, *security guidance for critical areas of focus in cloud computing V3.0*, Cloud Security Alliance, 2011

[2] Worku, S. G., Xu, C., Zhao, J., & He, X. Secure and efficient privacy-preserving public auditing scheme for cloud storage. Computers & Electrical Engineering 40, (2014), 1703-1713.

[3] Wang C,Wang Q, Ren K, Lou W. Privacy-preserving public auditing for data storage security in cloud computing. In: INFOCOM, 2010 proceedings IEEE; 2010. p. 1–9.

[4] Wang C, Chow S, Wang Q, Ren K, Lou W. Privacy-preserving public auditing for secure cloud storage. IEEE Trans Comput 2011

[5] Ateniese G, Burns R, Curtmola R, Herring J, Khan O, Kissner L, et al. Remote data checking

using provable data possession. ACM Trans Inf Syst Secur2011;14:1–34.

[6] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, et al. Provable data possession at untrusted stores. In: Proceedings of the 14th ACM conference on computer and communications security, CCS 2007. p. 598–609.

[7] Bowers KD, Juels A, Oprea A. HAIL: a high-availability and integrity layer for cloud storage. In: Proceedings of the 16th ACM conference on computer and communications security, CCS 2009. p. 187–98.

[8] Deswarte Y, Quisquater J-J, Saïdane A. In: Jajodia S, Strous L, editors. Remote integrity checking: integrity and internal control in information systems VI, vol. 140. Boston: Springer; 2004. p. 1–11.

[9] Dodis Y, Vadhan S, Wichs D. Proofs of retrievability via hardness amplification. In: Reingold O, editor. Theory of cryptography, vol. 5444. Berlin/Heidelberg: Springer; 2009. p. 109–27.

[10] Erway C, Kupcu A, Papamanthou C, Tamassia R. Dynamic provable data possession. In: Proceedings of the 16th ACM conference on computer and communications security, CCS 2009. p. 213–22.

[11] Shacham H, Waters B. Compact proofs of retrievability. In: Pieprzyk J, editor. Advances in cryptology – ASIACRYPT 2008, vol. 5350. Berlin/Heidelberg: Springer; 2008. p. 90–107.

[12] Zheng Q, Xu S. Secure and efficient proof of storage with deduplication. In: Proceedings of the second ACM conference on data and application security and privacy, CODASPY 2012. p. 1–12.

[13] Zheng Q, Xu S. Fair and dynamic proofs of retrievability. In: Proceedings of the first ACM conference on data and application security and privacy, CODASPY 2011. p. 237–48.

[14] Wang Q, Wang C, Li J, Ren K, Lou W. Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes M, Ning P, editors. Computer security – ESORICS 2009, vol. 5789. Berlin/Heidelberg: Springer; 2009. p. 355–70.

[15] Zhuo H, Sheng Z, Nenghai Y. A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. IEEE Trans Knowl Data Eng 2011;23:1432–7.

[16] Chunxiang X, Xiaohu H, Daniel-Abraha W. Cryptanalysis of Wang's auditing protocol for data storage security in cloud computing. In: Liu C et al., editors. Information computing and applications, vol. 308. Berlin Heidelberg: Springer; 2012. p. 422–8.

[17] Wang Q,Wang C, Ren K, LouW, Li J. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 2011;22:847–59.

[18] Liu C, Yang C, Zhang X, Chen J. External integrity verification for outsourced big data in cloud and IoT: A big picture. Future Generation Computer Systems 49, (2015), 58-67.

[19] Zhang J, Dong Q. Efficient ID-based public auditing for the outsourced data in cloud storage. Information Sciences 343-344, (2016), 1-14.

[20] Yang G, Yu J, Shen W, Su Q, Fu Z, Hao R. Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. The Journal of Systems and Software 113, (2016), 130-139.

[21] Yu Y, Niu L, Yang G, Mu Y, Susilo W. On the security of auditing mechanisms for secure cloud

storage. Future Generation Computer Systems 30, (2014), 127–132.

[22] Yu Y, Ni J, Au M-H, Liu H, Wang H, Xu C. Improved security of a dynamic remote data possession checking protocol for cloud storage. Expert Systems with Applications 41, (2014), 7789–7796.

Design and Implementation of on-board satellite encryption with SEU error detection & correction code on FPGA

Samah Mohamed, Khaled A.Shehata, Hanady H.Issa *(Arab Academy For Science & Technology (AAST)*

Nabil Hamdy Shaker *(Misr International University (MIU))*

Abstract

Earth Observation (EO) satellites in Low Earth Orbit (LEO) provide earth with data required for both military and civilian applications. Satellite manufactures are realizing that security is essential issue in satellite communications. Advanced Encryption Standard (AES) is one of the important candidates to secure satellite communications. Harsh radiation is the main feature of LEO environment which causes Single Event Upsets (SEUs). On-board encryption processor needs to be robust enough for faults in order to avoid transmission of erroneous data to ground. The presented algorithm in this paper combines AES with Hamming error detection and correction code to protect the on-board encryption process from SEU. The proposed fault tolerant algorithm is designed using the Hardware Description Language (HDL) design entry and implemented on Xilinx Field Programmable Gate Arrays (FPGAs) virtex6.

Keywords- AES, SEU, Hamming Error Detection and Correction Code, VHDL, FPGA.

i. Introduction and background

The science of cryptography started from the times of ancient Egyptians till today and its importance is increasing day by day. In recent years, with the explosive advancement in technology, the need for data security becomes essential especially in free space satellite applications [1].

AES is considered as security candidate well suited for resource constrained satellites platforms because of its simplicity, flexibility, ease of implementation and high throughput. In order to meet the requirement for high data rate processing demanded by present EO satellites, hardware implementation is considered as the preferred choice in satellites imaging payloads [2,3].

Field Programmable Gate Arrays (FPGAs) is well suited platform because of its flexibility of design, shorter time-to-market, remote configurability etc. So it is suitable for use in small satellite on-board systems. Increasing the immunity of encryption process against faults is very essential in satellites [4]. The induced faults due to harsh environments cause SEU which leads to erroneous data transmitted to ground. SEU fault detection is not enough for space applications but fault correction is more important [5].

Realizing security through various Encryption Algorithms and solving the SEU problem in satellite have been addressed in many references. In [3,6], combining of AES with Hamming Code (12,8) was presented without any detailed design. They implemented the system on Virtex2 (XCV2V1000) from XILINX with a maximum frequency of 25MHz. In [7] combining of Data Encryption Standard (DES) and Turbo code was presented without any hardware implementation. The main weaknesses of the system were the low level of security and small data block size as a result of using DES Algorithm. In [8] combining AES with Hamming code (12,8) was also presented but without any hardware implementation.

This paper presents combining both Hamming error detection and correction code with AES in a single algorithm. The algorithm is implemented on an FPGA and tested against SEU by injecting random faults.

ii- The System Architecture Of the Proposed Algorithm

This section describes a new approach of fault tolerant model. It combines both AES 128-bit block cipher and (12,8) Hamming code. The AES 128-bit block cipher output has 128-bit plaintext input and 128-bit session key as present in figure 1.

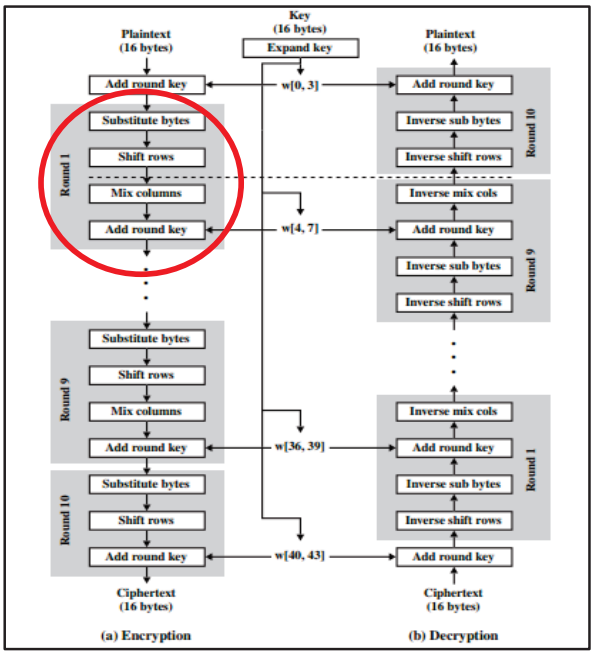


Figure 1. AES 10 Rounds Block Diagram

The used AES has 10 rounds, each round composes of four transformations named, Subbytes, Shift row, Mix column and Add round key.

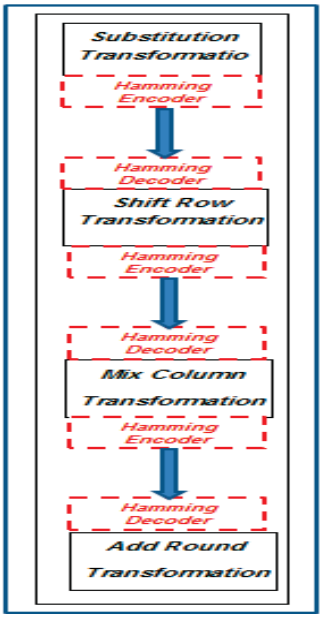


Figure 2. One round of AES combined with Hamming

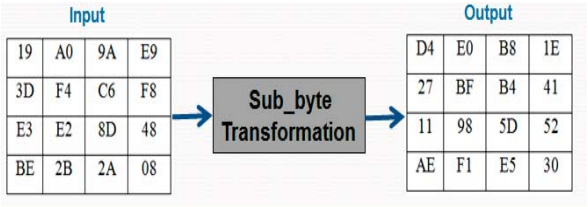
Figure 2 shows one round architecture of the proposed design. Hamming code (12,8) is inserted in the path of AES as an error detection and correction. It is added between transformations interconnects and is implemented 16 times in parallel in each 16 bytes state matrix [9]. Hamming error detection and correction operates on byte level in each AES state transformation matrix of all rounds [2,3,4]. Hamming encoding is performed after each transformation to get 192 bits while decoding is performed before the next transformation to get the corrected 128 bits (16 bytes) due to any SEU in its bytes. The main purpose of the combining AES with Hamming is to make sure that data transferred between two successive encryption transformations is error free, with no negative effect on the encryption process [3,4,8,9].

iii. AES transformation module design.

The system is designed using Matlab, before starting the hardware implementation, for verification purposes. For hardware implementation the Hardware Description Language (VHDL) design entry is used for the whole system design. Both the pre-routing and post-routing simulations of the proposed design are then performed using Modelsim 6.3a tools from Mentor Graphics. The following subsections show the pre-routing simulation of each AES transformation individually.

Subbytes transformation

Figure 3 presents the pre-routing simulation of subbytes transformation. The results are compared with Matlab results to indicate the proper operation of the subbyte transformation based on S-Box table. The S-Box contains a permutation of all possible 256 of 8-bit values. In this transformation each individual byte of state matrix is mapped into a new byte.



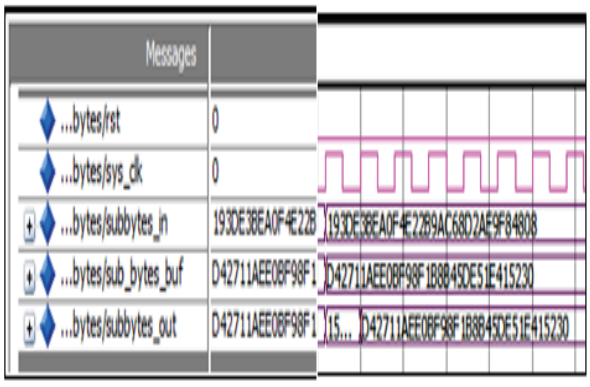
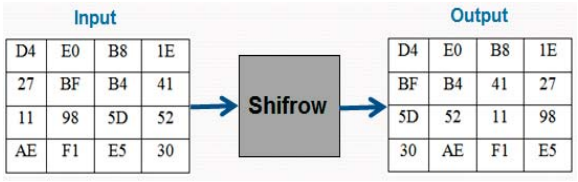


Figure 3. Subbytes pre-routing simulation of first Round

Shiftrow transformation

The input of this transformation is the output from the subbytes transformation. ShiftRows transformation causes diffusion of bits over multiple rounds by cyclically shifts the rows of the state over different offsets. Row number 0 in the matrix is not shifted, row 1 is shifted left by one byte, row 2 is shifted left by two bytes, and row 3 is shifted left by three bytes constructing the new state matrix.

Figure 4 shows that the simulation results of the transformation outputs are identical to those from Matlab. This indicates the proper operation of the shiftrow transformation.



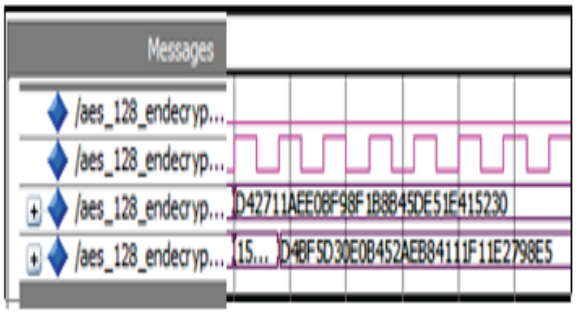
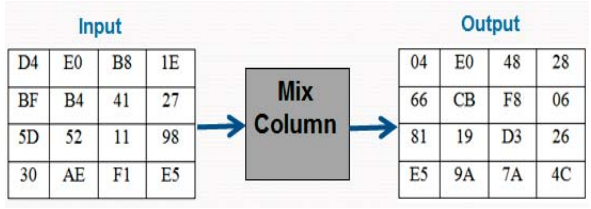


Figure 4. Shiftrow pre-routing simulation of first round

Mixcolumn transformation

The input of this transformation is the output from the shiftrow transformation. This linear transformation operates on the state matrix column by column. The matrix obtained from the ShiftRow step i.e. $[C_{ij}]$, is multiplied by a standard matrix to produce a new output matrix $[d_{ij}]$.

Figure 5 shows a coincidence between Matlab input /output results and pre-routing simulation result of this transformation.



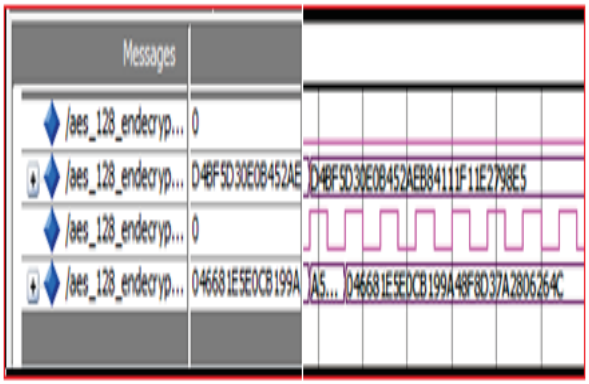


Figure 5. Mixcolumn pre-routing simulation of first round

AddRound Key

Figure 6 presents the pre-routing simulation based on XORing between the Mixcolumn output state matrix and key expansion of each specific round.

Mixcolumn					Round Key			
04	E0	48	28	XOR	A0	88	23	2A
66	CB	F8	06		FA	54	A3	6C
81	19	D3	26		FE	2C	39	76
E5	9A	7A	4C		17	B1	39	05

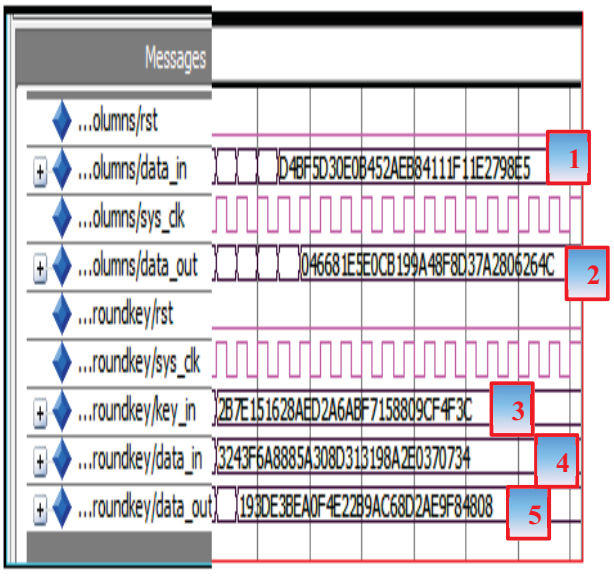


Figure 6. AddRound Key pre-routing simulation of first round

In figure 6, Number 1 shows the ShiftRow output state matrix in the first AES round. Number2 in the same figure indicates Mixcolumn output state matrix in first AES round. Number3 indicates Input key state matrix. Number 4 shows input plaintext state matrix. Number 5 displays the final output state matrix resulted from XORing mixcolumn output with the the first round key expansion output.

iv. Simulation Results of the Proposed System

This section concerns both the post and pre-routing simulations of the proposed design. Figure 7 shows the final cipher output of the proposed system as a pre-routing simulation.

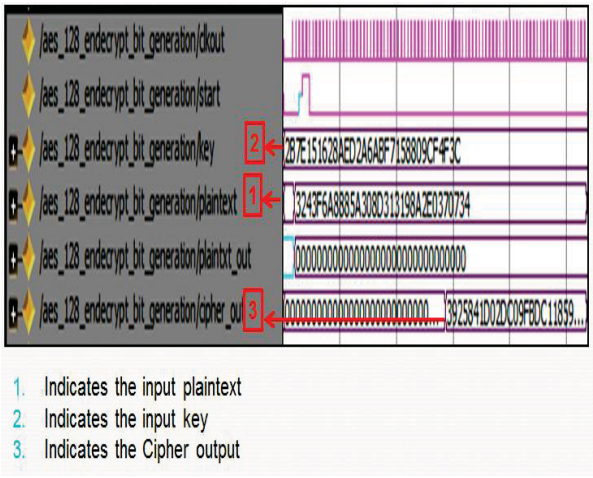


Figure 7. Pre-routing simulation results of fault free system

As shown in figure 7 the plaintext "plaintext_in" is $(3243f6a8885a308d313198a2e0370734)_h$. The key input "key in" is$(2b7e151628aed2a6abf7158809cf4f3c)_h$. The fault free output of the AES encryption process "cipher output" is named "plaintext_out" which is $(3925841d02dc09fbdc118597196a0b32)_h$. The corrected output with a single bit injected error in each byte is noted as "plaintext_out" which is: $(3925841d02dc09fbdc118597196a0b32)_h$.

The injection of faults is forced inside the different AES rounds at different transformation state matrix of any round. The corrected output is identical to the expected output. This means the system detects all single-bit-error in any byte and corrects them during the encryption process.

Figure 8 shows the post-routing simulation of the proposed system. The maximum delay of the output is 15ns which enable the system to run at 66MHz.

Figure 8 Propose system post routing simulation 4 cycles / round

Figure 9 shows the whole encryption process containing the 10 rounds. The whole operation lasts 40 clock cycles. The system is pipelined which allows the inputs to be applied in each clock cycle. This figure shows that the design corrects any SEU fault in different rounds and different transformations inside the round.

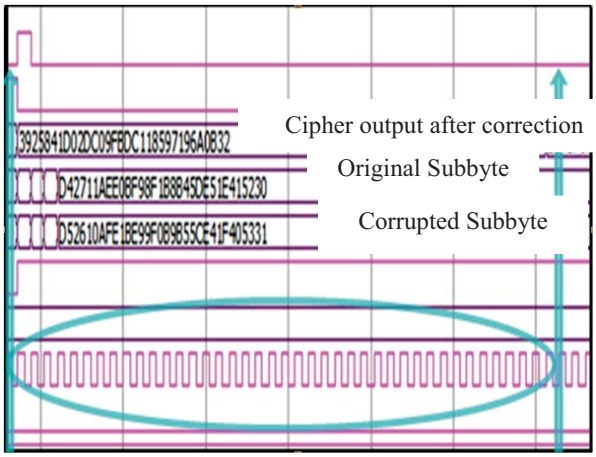


Figure 9 Propose System post routing simulation with error correction

v. Proposed System Faults Injection

The proposed system is simulated and tested through injecting faults randomly in different rounds, transformations and bytes. The main constrain is in injection of a maximum 1 bit fault in each byte to get proper error correction. In each simulation, the error is corrected as long as injecting only one bit-error in each byte. All results after fault correction are verified with Matlab simulation results [9]. The system corrects all cases of fault injection.

Figure 10 shows the injection of faults in three interconnect locations between AES transformations simultaneously; keeping in our consideration injecting of only 1 bit fault in any byte. The first interconnect location is between output of subbyte transformation and input of shiftrow presented by letter (A). The second interconnect

location is between output of shiftrow and input of Mixcolumn transformation presented by letter (B). The third interconnect location is between output of Mixcolumn and input of AddRound Key transformation presented by the letter (C).

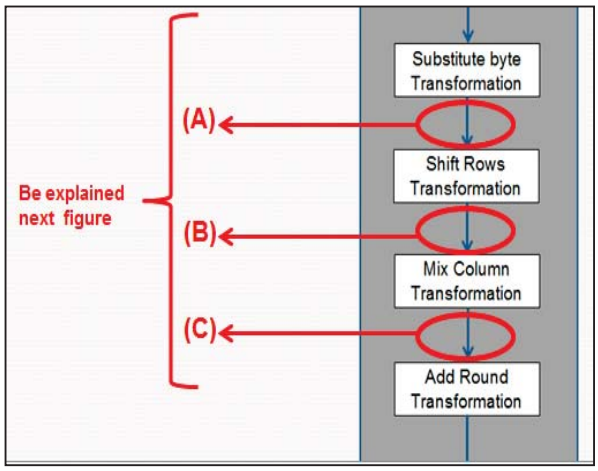


Figure 10 Faults injection in Second Round of AES Transformations

Figure 11 shows the three faulty matrices in the above interconnects which are indicated by letters A,B and C. The circled digits indicate the fault existence as shown in figure 10.

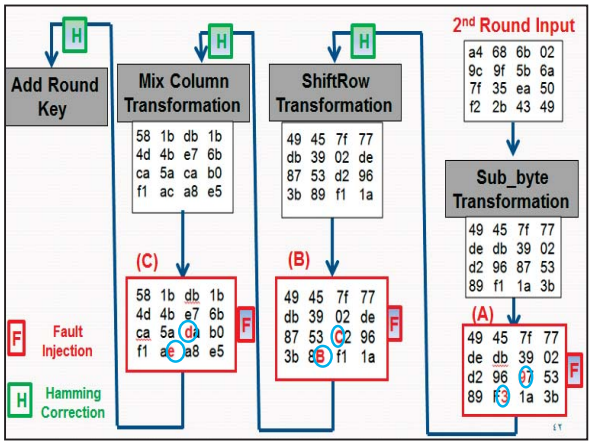


Figure 11: Inject 1 bit fault in a byte in more than one transformation in Second Round

Letter (A) in figure 11 shows the 2nd Round Subbytes matrix after fault injection as displayed in figure 12. Faults injected in specific matrix locations are as follows:-

- In bit no.(7) of byte no.(8)
- In bit no.(4) of byte no.(11)

Lower table in figure 12 indicates the injection of faults in 192 bits. These faulty bits are circled.

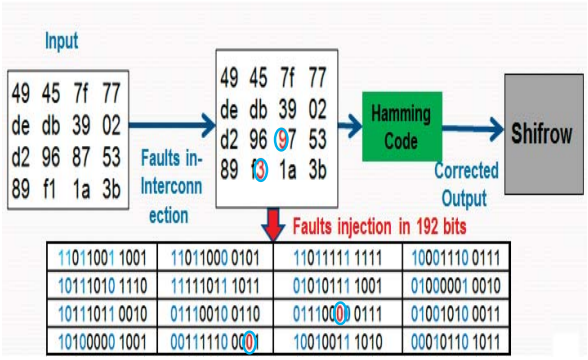


Figure 12: Part (A) Detailed explanation from Figure 11

Letter (B) in figure 11 shows the 2nd Round ShiftRow matrix after fault injection as displayed in figure 13. Faults injected in specific matrix locations as follows:-

- In bit no.(7) of byte no.(8)
- In bit no.(4) of byte no.(11)

The table in figure 13 indicates the injection of faults in 192 bits simultinously, the faulty bits are circled.

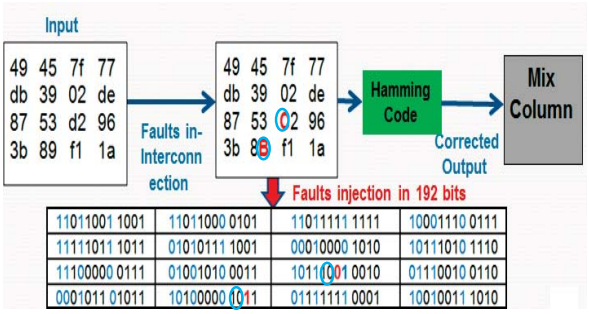


Figure 13: Part (B) Detailed explanation from Figure 11

Letter (C) in figure 11 shows the 2nd Round Mixcolumn matrix after fault injection as displayed in figure 14. Faults injected in specific matrix locations as follows:-

- In bit no.(7) of byte no.(8)
- In bit no.(4) of byte no.(11)

The table in figure 14 indicates the injection of faults in 192 bits, the faulty bits are circled. In all cases the faults are detected and corrected.

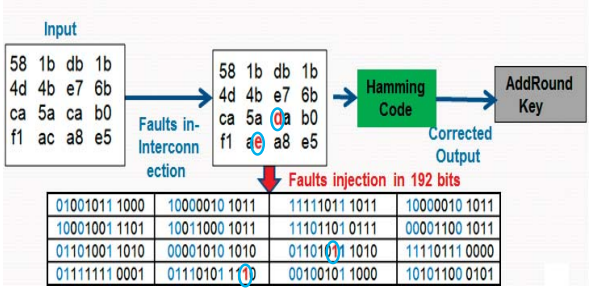


Figure 14: Part (C) Detailed explanation of Figure 11

vi. FPGA implementation of the proposed system

AES is efficient for hardware based implementation to meet the requirement of high throughput [9,10]. In most EO satellites high throughput fault – tolerant encryption process is required to satisfy high data rate transmission.

The VHDL is used as a design entry for the hardware design. Synthesis is carried out using Xilinx ISE tool version13. It generates a map reports which shows the FPGA utilization of the proposed design. The target FPGA used for hardware implementation is Xilinx XC6VLX240 – IFFG1156 and the FPGA Evaluation kit model ML605 is used. The hardware measurement is displayed on PC after the end of each encryption and decryption process to verify the correct output data.

The throughput of the fault tolerant AES implementation is calculated using the equation below:

$$\text{Throughput} = \left(\frac{128}{n}\right) \cdot f$$

where n is the number of clock cycles required to encrypt a single block of 128 bits. The system needs 4 clock cycles for each round and extra 2 clock cycles to prepare the session key. Accordingly, the total clock cycles n is (4*10)+2=42 clock cycles. The maximum system operating frequency f is 66MHz. So, the calculated throughput is equal to 201Mbps. But, due to pipelining of our designed system, which allows applying of 128 bits every clock cycle, the calculated throughput is 128 bit * 66MHz = 8.448 Gb/s.

Table1 presents the utilization of vertix6 FPGA.

Table 1. FPGA Utilization Report

Target FPGA is Xilinx XC6VLX240 & f= 66MHz			
Item	Used	Available	Utilization
Occupied slices	1,150	37,680	3%
IOB's	11	600	1%
Peak Memory usage	680 MB		
No. of LUT Slice	2,375	150,720	1%

vii. Conclusion

In this paper, we propose a hardware design for fault tolerant system combining AES with Hamming error detection and correction code. It protects the on-board satellites data during encryption process against SEU. The proposed design is implemented on Vertix6 FPGA. The utilization is 3% of the chip. The maximum operating speed is 66MHz with a throughput of 8.448 Gb/s which satisfies the satellite requirements. Varies Harsh environment simulations are performed for test purposes, which simulates the LEO Harsh environment. The proposed system detects and corrects a one SEU in each data byte of state matrix.

viii. References

[1] Pradeep Kumar Singh and Prof. Dipti Patil, "Comparative Study of Satellite Image Encryption Algorithm", International Journal of Infinite Innovations in Technology (IJIIT), Volume-I, Issue-II,Paper-08, India, pp. 1-23, October 2012.

[2] Ashkan Masoomi and Roozbeh Hamzehiyan," A New Approach for Detecting and Correcting Errors in the Satellite Communications Based on Hamming Error Correcting Code", International Journal of Computer Theory and Engineering(IJCTE), Vol.5, No.2, Iran, pp. 227-231, April 2013.

[3] Roohi Banu and Tanya Vladimirova,"Fault-Tolerant Encryption for Space Applications", IEEE Conference of Electronic Engineer, Surrey Space Center (SSC), VOL.45, NO.1, India & Russia, pp. 266-279,January 2009.

[4] Nahid Farhady Ghalaty, Aydin Aysu and Patrick Schaumont, "Analyzing and Eliminating the Causes of Fault Sensitivity Analysis",IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE), USA, PP.1-6, 2014.

[5] Hoda Pahlevanzadeh, Jaya Dofe, and Qiaoyan Yu, "Assessing CPA Resistance of AES with Different Fault Tolerance Mechanisms", IEEE conference of 21st Asia and South Pacific Design Automation Conference (ASP-DAC),USA, pp.661-666, 2016.

[6] Roohi Banu and Tanya Vladimirova, "On-Board Encryption in Earth Observation Small Satellites " ,IEEE Conference of Electronic Engineering , Surrey Space Center (SSC),United Kingdom, pp.203 – 208, 16-19 October 2006.

[7] Rajashri Khanai , Dr. G. H. Kulkarni and Dattaprasad A.Torse , " Neural Crpto-Coding as DES : Turbo over Land Mobil Satellite (LMS) Channel" , IEEE, International Conference in Communications and Signal Processing(ICCSP), India, pp.1300-1305, 3-5 April 2014.

[8] C. Jeba Nega Cheltha and Prof. R.Velayutham , " A novel Error-Tolerant Method in AES for Satellite Images", IEEE International Conference in Emerging Trends in Electrical and Computer Technology (ICETECT),India, pp. 937-940 , 23-24 March 2011.

[9] Samah Mohamed, KhaledA.Shehata, HanadyH.Issa and Nabil Hamdy Shaker " FPGA Implementation of a combined Hamming – AES error tolerant algorithm for on board satellite ",IEEE Conference of Electronics Engineer, The World Congress on Information Technology and Computer Applications (WCITCA'2015), Hammamet, Tunisia, pp. 1-4, 11-13 June 2015.

[10] C.Thamilarasi and K.Shanmugapriya, "A HIGH THROUGHPUT AND ERROR TOLERANT AES DESIGN ",International Journal of Advanced Research in in Electronics and Communication Engineering (IJARECE), Volume 2, Issue 4, pp.420-424, April 2013.

ISBN: 1-60132-445-6, CSREA Press ©

SESSION

SECURITY APPLICATIONS

Chair(s)

Dr. Greg Vert

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

Strategic Risk Management in Counter-Terrorism for the Railbound Public Transport

Merging Qualitative and Quantitative Operations Research Techniques

Martin Zsifkovits*, Stefan Pickl
Universität der Bundeswehr München
Institute for Informatics, Mathematics, and Operations Research
Neubiberg, Germany
*martin.zsifkovits@unibw.de, stefan.pickl@unibw.de
*Corresponding author

***Abstract*—Every modern state is strongly dependent on a functioning infrastructure. This makes it even more vulnerable and furthermore attractive for terroristic attacks. The situation gets even more severe when people are directly involved, such as in public transport, as they are – at least for some groups of terrorists – the main aim of attacks. In the paper at hand we propose the standardized ISO31000 risk management framework coupled with various qualitative and quantitative Operations Research techniques in order to tackle the strategic risk management for counter terrorism. The framework is applied to the case of public rail bound transport and illustrated on research that was conducted in the sponsored project RiKoV.**

***Keywords*—Counterterrorism, Strategic Risk Management, Rail Security, Operations Research, Reachback**

***Track*—Security Management**

I. Introduction

Society depends decisively on the availability of infrastructures such as energy, telecommunication, transportation, banking and finance, health care and governmental and public administration. Even selective disruption of one of these infrastructures may result in disruptions of governmental, industrial or public functions; in general in public management. Vulnerability of infrastructures therefore offers spectacular leverage for natural disasters as well as criminal actions. Threats and risks are part of the technological, economical, and societal development. Increasing complexity of our critical infrastructures exacerbates consequences of natural and/or man-made disasters. Not only primary effects but also cascading effects as result of increasing dependencies and interdependencies of our technological and societal systems demand intelligent simulation and optimization techniques in the area of operations research, system dynamics and public management:

A comprehensive safety and security management should be part of a modern public management.

In this context, terroristic threats are getting more and more important in global politics and managerial decision making. Especially providers of critical infrastructures need to handle those threats and prepare themselves for a huge variety of possible events and threats. One of the main reasons for this is that critical infrastructures, such as public transport, are open systems and therefore difficult to protect [1], as well as prestigious for terroristic attacks. Hence, a structured risk management process is needed in order to handle the complex situation and evaluate different risks.

In prior publications with colleagues we proposed the risk management process according to ISO31000 [2] for strategic foresight [3], national security [4], and cyber threats to critical infrastructures [5]. In the paper at hand we demonstrate the application of this standardized risk management process to counter-terrorism in the rail bound public transport sector. The terroristic threat on this critical infrastructure in general was focus of our project on the costs and risks of terroristic threat on the rail bound transport – RiKoV, sponsored by the German Ministry of Education and Research (BMBF) [6]. Now, after several analyses have been made and tools created, we aim at merging all approaches into an overall integrated management framework.

The paper is structured as follows: in section 2 we show the management framework based on ISO31000, where we explain the steps of the central stages of risk assessment, risk identification, risk analysis, and risk evaluation, in detail. Afterwards we give some insights into the findings of the approach applied to RiKoV in section 3 and finally conclude the paper in section 4.

II. Management Framework

The overall aim of strategic risk management is the minimization of risks for humans and their livelihoods, or at least keeping it within acceptable bounds. This makes the risk management process even more complex and demands for a holistic perspective [7]. Therefore, the standardized risk management process ISO31000 is of great value for decision makers, as it allows for a structured management process and leads to transparent decision making.

The risk assessment is put into a framework that starts with the establishment of the context. The context defines the strategic planning horizon, the points of interest and other basic

elements. While the risk assessment, monitoring and review as well as communication and consultation ensure possibilities for reach back and allow for real-time validation. The aim of the process is adequate risk treatment, which again influences the establishment of the context in a next step.

In the paper at hand we concentrate on the risk assessment, as it is the most critical part of the risk management for strategic decisions. Figure 1 shows its main elements, which are the identification of risks, their analysis and the evaluation.

Figure 1: Risk Assessment

The results and findings of each step are the basis for further steps, meaning that the process needs to be gone through sequentially.

A. Risk Identification

The first stage of the risk assessment, the risk identification can be seen as the most critical step in the process, as all following analyses and evaluations are based on the quality of identified risks. Thus, the resulting risk treatment can only be as good as the identified risks [8]. Therefore we propose a multilayered approach for identifying pending risks. In the context of terroristic threats on the rail bound public transport we started with a historical analysis of terroristic attacks in order to identify terrorists and their aims, used weapons, routes of attacks and the corresponding target. In addition to this, literature was screened and reviewed for getting further insights. We identified propaganda magazines and blogs as especially insightful, as they directly address attackers and discuss the most relevant issues – also for counter-terrorism. In contrast to some of our previous work (e.g. [3, 4], literature was screened and reviewed manually instead of with automated web crawling software due to the high degree of complexity of the texts and their messages. In a further step of identification, several experts on the field were interviewed and asked for their expertise, expectations and forecasts, and opinions. All findings were collected and in various focus groups discussed with scientist and experts on the field. These discussions resulted in a final listing of hazards. The overall process of the risk identification phase is shown in Figure 2.

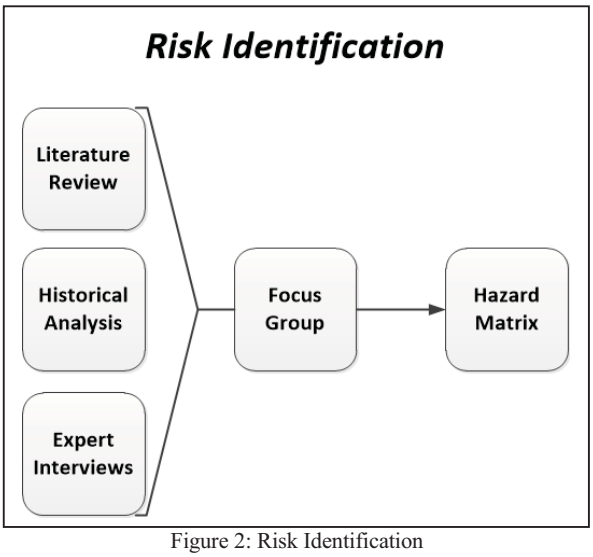
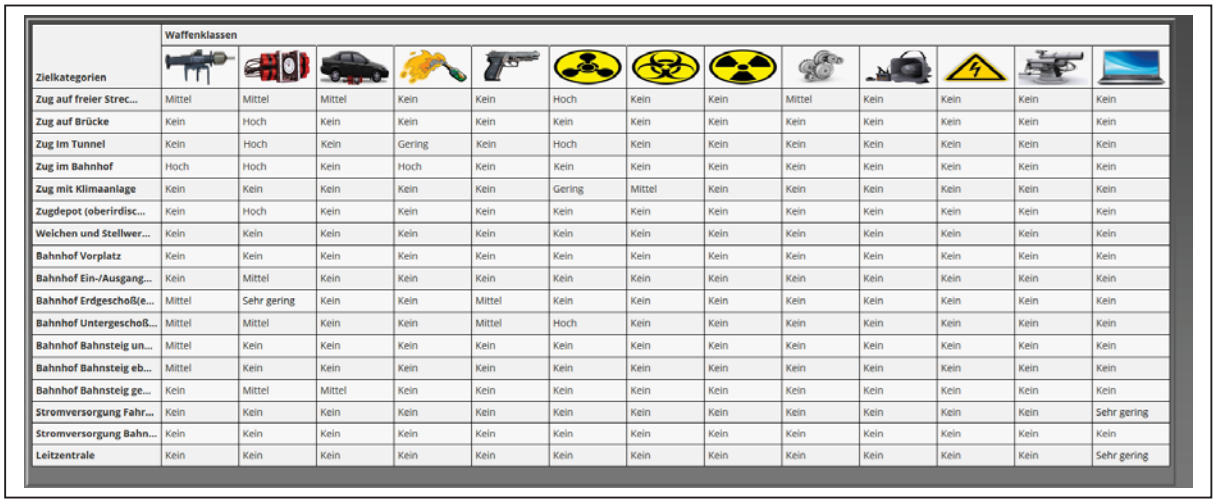


Figure 2: Risk Identification

As the identified hazards were too complex for a simple "catalogue of hazards", an extensive matrix was designed that considers weapons and targets, as well as their interplay. For the representation a database was set up and implemented in a visualization platform. Here, the matrix itself can be illustrated in terms of consequences, hazards, damages or risks. The risks-illustration is shown in Figure 3. Even though the matrix is in German, the logic is rather easy to understand. In the rows several scenarios that were analyzed before are described, such as "train on main track", "train on bridge", "train in tunnel", etc. The columns represent weapons, such as machine guns, explosive, car bomb, incendiary composition, gun, and others. The matrix is based on fuzzy logic analysis and groups the outcomes for all matrix types into very low (sehr niedrig), low (niedrig), medium (mittel), high (hoch), and very high (sehr hoch). The definition of groups is predefined by the decision maker and in our case based on experts' opinions.



Zielkategorien	Waffenklassen												
Zug auf freier Strec...	Mittel	Mittel	Mittel	Kein	Kein	Hoch	Kein	Kein	Mittel	Kein	Kein	Kein	Kein
Zug auf Brücke	Kein	Hoch	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Zug im Tunnel	Kein	Hoch	Kein	Gering	Kein	Hoch	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Zug im Bahnhof	Hoch	Hoch	Kein	Hoch	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Zug mit Klimaanlage	Kein	Kein	Kein	Kein	Kein	Gering	Mittel	Kein	Kein	Kein	Kein	Kein	Kein
Zugdepot (oberirdisc...	Kein	Hoch	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Weichen und Stellwer...	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Vorplatz	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Ein-/Ausgang...	Kein	Mittel	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Erdgeschoß(e...	Mittel	Sehr gering	Kein	Kein	Mittel	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Untergeschoß...	Mittel	Mittel	Kein	Kein	Mittel	Hoch	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Bahnsteig un...	Mittel	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Bahnsteig eb...	Mittel	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Bahnhof Bahnsteig ge...	Kein	Mittel	Mittel	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Stromversorgung Fahr...	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Sehr gering
Stromversorgung Bahn...	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein
Leitzentrale	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Kein	Sehr gering

Figure 3: Hazard Matrix

The risk identification phase gives deeper insight into existing risks and threats and can also be used to test the effectiveness of possible security measures and installations.

B. Risk Analysis

Based on the prior identifications, the sum of all hazards is analyzed in the next step of the risk assessment. To this end we again combine several methods in order to reduce shortcomings of individual techniques and, at the same time, gain from the multiple benefits. The methods and their interconnections are illustrated in Figure 4.

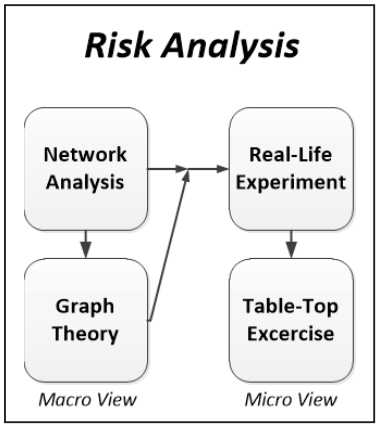


Figure 4: Risk Analysis

The risk analysis was split in an analysis on the macro level, detecting and analyzing threats on a global level, and an analysis on the micro level, on which individual threats within train stations were identified.

Macro View

The aim of the analysis on the macro level was identifying train stations with the highest vulnerability. We assume, that those station might be more attractive to attackers than others, as they are characterized by a higher number of train-through traffic, more passengers per day, a central location in the system, or others. Therefore, several different analyses were conducted.

In a first step, the underlying network, in our case the German train network, was illustrated and set up for analysis. Several measures were underlied, such as the distance between stations, the number of trains going between them, the number of passengers per station, and others.

Graph theoretic measures were performed in order to analyze the vulnerability of the network. Mainly measures from biology and chemistry were applied, where one measure was adapted in order to compare the distances between nodes and their efficiency in terms of train and passenger flows, the flow-weighted efficiency measure (see [9] for further information). In order to identify the critical stations, also four centrality measures: the degree centrality, closeness centrality, eigenvector centrality and betweenness centrality were used [10, 11]. For the sake of analyzing the robustness, seven kinds of vulnerability measures were applied: vertex connectivity measure, vertex toughness measure, vertex scattering number measure, vertex integrity measure, vertex tenacity measure, vertex domination number measure and edge multi-scale measure based-edge-betweenness-centrality [12-14]. Figure 5 shows an exemplary result of the graph theoretic analysis, the illustration of the closeness centrality measure.

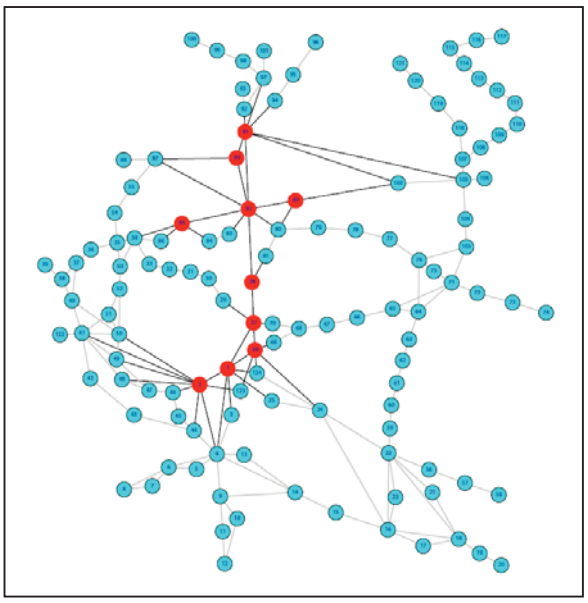


Figure 5: Closeness Centrality on German Railway System

Micro View

After detecting the most critical stations, the second phase on the micro view analyses threats within those stations and tries to evaluate security measures and installations. Two events were undertaken in order to evaluate the status quo in the microscopic perspective, a real-life experiment and a table-top exercise.

Within the RiKoV-project, a real-life experiment was executed in May 2015 in Cologne, Germany. Therefore, an underground station was used and about 100 actors performed a daily situation in a train station. Additionally, several people were set as terrorists, carrying hidden weapons with them. In several scenarios and runs, different security installations (such as distance based sensors, camera systems, body checks, etc.) were tested on their impact in different settings. The potential hotspots within such a station were identified and the impact of security installations in several attack scenarios were tested. One main question that should be answered was, if the system could be transferred to a closed system with security checks, respectively which security installations should be preferred. For more information on this experiment see [1, 15].

In a further step, experts were invited to a table-top exercise that was conducted directly after the experiment by the TH Köln. The scenarios and their outcomes where discussed based on a 3D representation of the station, in which the experiment was conducted in. This allowed experts to bring their experience into the results and reduced experiment errors in the overall results. The exercise also allowed the discussion of further scenarios and their range of possible outcomes.

C. Risk Evaluation

In the final stage of the risk assessment, we tried to evaluate all the findings from the prior steps. As the main interest was in getting insights into single attacks and individual protection of train stations, we decided to use Agent Based Modeling. The methodology has gained considerable importance in the last decade due to the increases in computational power. It allows for modeling individuals and their behavior and the aggregation on a more macroscopic level. In the evaluation at hand this means that we model individuals in a train station based on findings in literature and observations (see [16]), as well as data from the real-life experiment.

In a first approach we aimed at identifying the most critical spots in terms of passenger density within Munich main station and conducted data farming experiments based on rather simple agent behavior. Results were represented in heat maps of the station, differentiating between the passengers' densities in the station and the durations of their stay at several spots (see [16]). Those heat maps also give insights into the optimal placement of security installations, especially as the system is not closed and installations need to be distance based. A resulting heat map for the main station Munich is shown in Figure 6.

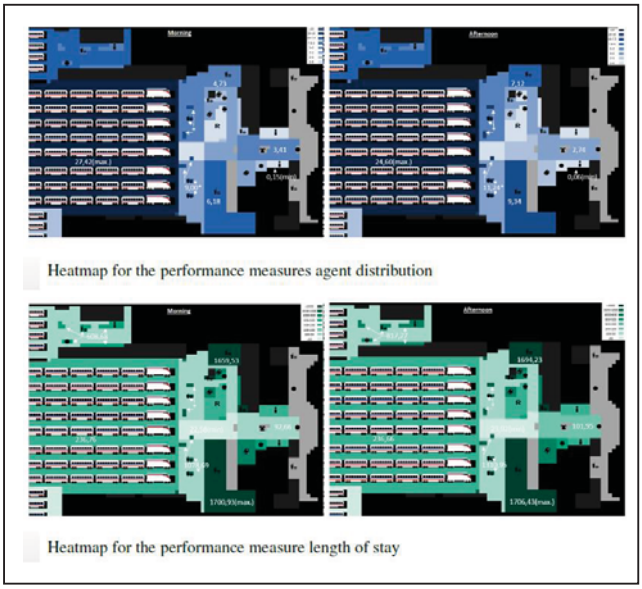


Figure 6: Heat maps from Agent Based data farming experiment [16]

Additionally, we aimed at evaluating concrete attack scenarios that were identified and analyzed before. Another Agent Based model was set up and parameterized with data from the local positioning system (LPS) from the experiment. We created the environment from the real-life experiment and were able to test for different attack strategies, multiple weapons, and game-theoretic attack paths.

For a realistic representation of the simulations and the clear and straight forward evaluation of security measures, we constructed a 3D environment using the software package *Cinema4D* and also used standard characters from this package. The simulation was then made using the game engine *Unity5.1*. A screenshot from a simulated security check including a distance based detector for explosives is shown in Figure 7, a screenshot of a simulated attack is shown in Figure 8.

Figure 7: Screenshot from Simulation - Security Check

The evaluation of security installations is based on parameters and findings from the experiment and allows individual tests on the topic. In the simulated attack from Figure 8, passengers are parameterized based on data from the experiment, whereby the attacker behaves based on findings from the risk identification phase.

Figure 8: Screenshot from Simulation - Terroristic Attack

One aspect that turned out to be highly interesting for decision makers is the fact that the simulation also allows consideration of the evacuation after the attack. This gives valuable insights into evacuation plans and barriers that might lead to problems. Figure 9 shows a screenshot of such an evacuation scenario.

Figure 9: Screenshot from Simulation – Evacuation

III. Results

The application of the risk management framework to terroristic threats led to meaningful insights and results. However, it is a difficult task to decide which findings should be published, and which should rather stay internal in the project, as they might be abused. Hence, there are only several results that will be discussed in this section.

In the risk identification we realized a slight shift in the choice of weapons. Even if in historic attacks explosives were the main arms, especially the literature analysis implied a shift towards automatic guns. This assumption is further supported by the last attacks in Paris and in the Thalys train.

The risk analysis was a further step in understanding overall correlations and underlined, that the place and mode of an attack strongly rely on the attacker and his motivations. In the quantitative network analysis we could identify stations and connections that are at high risk for an attack with the main objective to disturb the system or stop its serviceability, while other measures gave insight into the criticality of nodes in terms of passenger appearance.

The real-life experiment combined with the table-top exercise clearly illustrated the advantages and disadvantages of single security technologies and additionally highlighted one major point: Public transport, as we know it today, cannot be converted to a closed system. Extensive security checks such as in aviation could not be handled in small stations and are not applicable to the large number of passengers. Additionally, the crowds caused by the waiting time at the checks might lead to a simple shift of the vulnerability and create new points of interest for possible attacks. Reduced checks might make sense if they do not disturb or influence the passenger flow. Thus, distance-based sensors have to be further developed in order to decrease the number of false alarms while increasing the detection rates.

The data farming experiment based on a first model on passenger behavior in a station with the resulting heat maps showed the potential of this approach and showed hot spots in a very concrete station. This analysis was later done based on different times (weekday morning, weekday afternoon, weekends, etc.) and showed the shifts in vulnerability over time. Thus, one needs to be aware of the point that public transport is a living environment and changing rapidly over time. Via the application of artificial intelligence in Agent Based modeling we showed that several security installations can reduce the vulnerability of a station, but attackers tend to

learn from systems and can find ways to overcome such barriers.

IV. Conclusion

Applying the risk management process based on ISO31000 allowed deeper insights into the topic and gave an idea of a structured architecture that might help decision makers, especially in the field of critical infrastructure protection, to identify-, analyze-, and evaluate pending risks. The prototypical match of several techniques and methods represents an excellent basis for implementation into an IT-based industrial framework. Especially the multilayered approach in the risk identification phase turned out to be very promising. However, one of the major shortcomings can be seen in the risk evaluation phase, as simulation models demand experts on the methodology and need individual input data and create results that cannot be necessarily compared with each other. A more automated approach would be fuzzy logic analysis that is currently under investigation for the phase of risk evaluation and future potentials. This might be integrated in a management cockpit that would be applicable for several fields of application, such as aviation, road transport, or the shipping industry.

Acknowledgment

The support from the German Federal Ministry of Education and Research (BMBF) (project RiKoV, Grant No.13N12304) is gratefully acknowledged. We also thank all the Partners in RiKoV (TH Köln, KIT, Airbus) for their valuable work that is partly also shown in this article.

References

[1] S. Meyer-Nieberg, M. Zsifkovits, S. Pickl, F. Brauner, "Assessing Passenger Flows and Security Measure Implementations in Public Transportation Sytsems", Proceedings of the Future Security Conference, Berlin, September 2015.

[2] International Organization for Standardization, "ISO31000: Risk Management – Guidelines for Principles and Implementation of Risk Management", 2009.

[3] M. Zsifkovits, S. Meyer-Nieberg, S. Pickl, „Operations Research for Risk Management in Strategic Foresight", GRF Davos Planet@Risk, vol. 3 (2), June 2015, pp. 281-288.

[4] M. Dehmer, S. Meyer-Nieberg, G. Mihelcic, S. Pickl, M. Zsifkovits, „Collaborative Risk Management for National Security and Strategic Foresight", EURO J Decis Process, vol 3 (3), November 2015, pp. 305-337.

[5] S. Meyer-Nieberg, M. Zsifkovits, "Cyber Threats: Introducing a Risk Management Framework for Cyber Security in Critical Infrastructure Protection", Proceedings of the Future Security Conference, Berlin, September 2015.

[6] S. Pickl, W. Raskob, A. Lechleuthner, W. Laible, W. Schmitz, et al. „Common Proposal for joint research project: RIKOV Risiken und Kosten der terroristischen Bedrohungen des schienengebundenen ÖPV: Eine Planungslösung für die ökonomische und organisatorische Optimierung präventiver und abwehrender Maßnahmen. BMBF Projekt Förderlinie - Sicherheitsökonomie und Sicherheitsarchitektur." 2011.

[7] Federal Office for Civil Protection, „Integrated Risk Management", Bern, Switzerland, 2014.

[8] S. Meyer-Nieberg, S. Pickl, M. Zsifkovits „Designing a Risk Management Framework for Forecasting National Security Issues", Safety and Security Engineering VI, 2015, pp. 39-48.

[9] S. M. Nistor, S. Pickl, M. Raap, M. Zsifkovits, "Network Efficiency and Vulnerability Analysis using the Flow-Weighted Efficiency Measure", Proceedings of the EMNET (Management and Governance of Networks) Conference, Cape Town, 2015.

[10] D. Gomez, J.R. Figueira, A. Eusebio, "Modeling centrality measures in social network analysis using bi-criteria network flow optimization problems", Europ J of OR, vol. 226(2), 2013, pp. 354-365.

[11] J. Wang, H. Mo, F. Wang, F. Jin, „Exploring the network structure and nodal centrality of china's air transport network: Acomplex network approach", J of Transp Geo, vol. 19(4), 2011, pp. 712-721.

[12] A. Mamut, E. Vumar "Vertex vulnerability parameters of kronecker products of complete graphs", Information Processing Letters, vol. 106(6), 2008, pp. 258-262.

[13] S. Alanko, S. Crevals, A. Isopoussu, P. Ostergard, V. Pettersson, "Computing the domination number of grid graphs", The Elec J of Comb, vol. 18(1), 2011, p. 141.

[14] S. Boccaletti, J. Buldu, R. Criado, J. Flores, V. Latora, J. Pello, M. Romance, "Multiscale vulnerability of complex networks", Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 17(4), 2007.

[15] F. Brauner, O.A. Mudimu, A. Lechleutner, A.Lotter, „Cologne Mass Casualty Incident Exercise 2015 - Evaluation by Use of Linked Databases to Improve Risk and Crisis Management in Critical Infrastructure Protection", Proceedings of the International Conference on Operations Research - OR2015, Vienna, 2015.

[16] S. Meyer-Nieberg, M. Zsifkovits, D. Hauschild, S. Luther, "Simulation-Based Analyses for Critical Infrastructure Protection: Identifying Risks by using Data Farming" Operations Research Proceedings 2015, 2015.

Vulnerabilities with Internet of Things

Dr. Kazi Zunnurhain
Assistant Professor
Northern Kentucky University
Highland Heights, KY 41099
+1 859 572 4753
zunnurhaik1@nku.edu

ABSTRACT
With the rapid growth of the Internet and rapid growth of smart device usage in our daily lives, we are submerged in an interconnected, ubiquitous and pervasive network, also known as the Internet of Things (IoT). We can see the use of IoT smart objects and devices in multiple settings: automobile industry, medical devices, industrial IoT, personal fitness apps, home appliances, and smart toys. Additionally, automation, central control, redundancy in service providing and tracking of IoT devices have reached the utmost feasibility to allure consumers toward IoT devices, particularly with the advancement of Cloud services and Big Data. Consumers can use IoT devices to keep track of commands, records the past sessions and reuse those commands for later to operate and manage the devices remotely. In these ways, IoT is gaining in popularity and also complexity. In the cybersecurity paradigm, securing IoT devices has become one of the newest and most daunting tasks.

In this paper, different aspects of security attacks have been considered for IoT devices. Vulnerabilities based on devices have been identified, with recent events in the cyber world as examples.

Categories and Subject Descriptors
D.3.3 [**Security**]: *Internet of Things.*

General Terms
Security.

Keywords
Internet of Things (IoT), vulnerability, cloud, security.

1. INTRODUCTION
According to Forbes Magazine, by 2020, the number of interconnected IoT objects and smart devices will exceed 50 billion. This emphasizes the need for security in IoT, considering the immense growth of IoT within next four years. Also, hosting of IoT objects and devices in cloud platform gets onboard with vulnerable issues from Cloud virtualization. Usually, with the development of an app to control an IoT device remotely, this is implemented in the cloud to undertake the services from a provider due to low cost and maintenance. However, the security measures of a public cloud or the app developing company is not always transparent to consumers. For example, consider an app that has been developed on a provider's cloud platform. The developer must have rented several virtual machines to run the app and consuming the cloud resources for his business. Now from a security point of view, it has to be ensured through Service Level Agreement (SLA) that the provider is completely trustworthy with all the personal credentials of the consumers who are using the app with complete trust to the app developing company or the hosting cloud. There should not be any possibility of man-in-the-middle attack between the consumer's home network and cloud network. Also the SLA (Service Level Agreement) should be transparent between the app developing company and the cloud provider, so that unusual phenomenon undergo proper investigation by a third party.

Provision of security standards for IoT objects should certainly maintain the security triads: Confidentiality, Integrity and Authentication. Consumers' information should be highly secure, provider and consumer should maintain a two-way handshake SLA to resolve any dispute. Finally, the access of the IoT device must be authenticated.

In Section 2, we will discuss some recent research on IoT security and existing vulnerabilities in their proposed model. In Section 3, we will highlight some recent attacks and security vulnerabilities in IoT objects and devices with a real-time example. In Section 4, we give an overview of both the physical and network vulnerabilities for IoT objects. Lastly, we will discuss the cloud involvement in IoT and different aspects of attacks with the engagement of IoT with cloud computing. Our goal is to highlight the most recent and prominent vulnerabilities in IoT devices and possibilities of external attacks from different aspects. We will also discuss the engagement of cloud computing platform for IoT devices from security aspects.

2. Related Work
There has been some research focusing on security issues with IoT objects and devices. Some related to Web IoT, and a few concerning Network protocol security. Additionally, there are very few research works focused on cloud engagement in IoT and proposed theoretical designs. In this section, some of those studies will be highlighted and discussed.

Díaz, M., Martín, C. and Rubio, B. [10] compared different cloud platforms, cloud infrastructures and IoT middleware. Also compared among different integration proposals and survey of data analysis techniques. But considering the varieties of IoT objects and devices to support from an orchestrated uniform platform was absent in their research work.

Weinberg, B.D., Milne, G.R., Andonova, Y.G. and Hajjat, F.M. [9] introduced the IoT to the broad managerial community and explored one of its central tensions: convenience vs privacy vs secrecy. They tried to clarify the ways in which IoT differs from web 2.0 and then highlighted opportunities, challenges and managerial guidance. Also depicted some prominent issues for privacy and secrecy. But their work lacked in introduction of

these prominent issues with cloud computing or any SOA to integrate with IoT devices or objects.

Sadeghi, A.R., Wachsmann, C. and Waidner, M. [3] emphasized on industrial IoT systems and related security and privacy challenges. Current trends and initiatives, such as "Industrie 4.0" and Internet of Things [22], [15] and [19], promise innovative business models and novel user experiences through strong connectivity and effective use of next generation of embedded devices. But did not mention the impact of attacks on a scalable industry where production growth is never finite. Also if the end product has a hardware fault, an adversary can use that for a Trojan attack and using that faulty part as a back door.

In the following sections we have highlighted different aspects and issues of vulnerabilities with recent example and tried to aware the security researcher to focus on specific issues.

3. VULNERABILITIES

This section discusses and highlights some of the most recent vulnerable issues with IoT security from different aspects in terms of usage. We will consider the automobile industry, medical devices, industrial IoT, home appliances, personal fitness apps, and smart toys.

3.1 Automobile Industry

Nowadays, we are fascinated by cars that come with high-end technologies, such as - Bluetooth capability, GPS mounted dashboard, automatic transmission, and remote start by Smart phones, and many more different apps preinstalled in car's dashboard monitor. All these technological advantages can cause serious vulnerabilities. Some of the scenarios will be depicted in this section.

Security researchers Charlie Miller and Chris Valasek forever transformed the automobile industry's notion of "vehicle safety" in July after demonstrating that they could remotely hack a 2014 Jeep Cherokee to disable its transmission and brakes [6]. Due to their work Fiat Chrysler had issued a first-time recall for 1.4 million vehicles, mailing out USB drives with a patch for the vulnerable systems and blocking the attack on the Sprint network that connected its cars and trucks [3].

In August at the DefCon hacker conference, Marc Rogers, security researcher for CloudFlare, and Kevin Mahaffey, CTO of mobile security firm - Lookout, revealed a suite of vulnerabilities they found in the Tesla Model S that would allow someone to connect their laptop to the car's network cable behind the dashboard. Enabling to start the engine of a $100,000 vehicle and drive off in it. They could also plant a remote-access Trojan on the car's internal network to later remotely cut off the engine while someone was driving [3], [6] and [22].

Also the same year at DefCon, a security engineer Samy Kamkar demonstrated his book-sized device, named OWNSTAR, to intercept a GM driver's OnStar smartphone app by planting his device in that car. This would give the adversary the ability to geolocate the vehicle, unlock it and remote start. That device works for BMW and Mercedes Benz app as well [13].

Another security issue was checked by University of California San Diego, when they exploit the dongle (a USB device used by a Car insurance company to keep track of car speed and acceleration) [17].

3.2 Medical Devices

Medical devices were not an exception from vulnerabilities with IoT as well. Some students in University of Alabama was able to hack into the Wi-Fi system of a Pacemaker implanted inside an iStan (a robotic dummy patient used to train medical students) [6]. The heart rate can be increased or decreased from a compromised Pacemaker by an external adversary. If the patient had a defibrillator, then it can be shocked repeatedly as well [6], and [7].

Another issue was with Drug Infusion Pump (usually patient in Intensive Care Unit) – to dole out morphine, chemotherapy and anti-biotic [2], [4]. Billy Rios, a security researcher, discovered serious security vulnerable points with the pump. Pump maker patched some vulnerabilities and claimed the rest issues are not vulnerable [6]. He also found security issues with insulin pumps.

3.3 Industrial IOT

Two nuclear power plants' monitoring systems were infected by a virus. It's called Slammer Worm. Slammer worm is a SQL worm if injected then it would cause a denial of service (DoS) attack. First it attacked on January 2003 and within 10 minutes 75000 victims were affected with that worm [5]. In computer security a worm is a malicious code segment that replicates itself and penetrates an operating system with ill intension, usually causing a denial of service attack or distributed denial of service attack. A worm engages the normal operational resources of an operating system and then legitimate operations/requests/queries starve. Hence resulting in denial of service (DoS). Another virus infected the signal and dispatching control system of a major transportation network in USA [6], [16] and [21].

3.4 Home Appliances

Household appliances are also vulnerable from different types of external attacks. Security breach on these IoT objects can take place in two ways. Either the system will be physically compromised or the central server monitoring the devices could be compromised. Some of the examples are listed below.

A Samsung "Smart Fridge" designed to sync over user's google calendar, failed to validate SSL certificates (SSL is a Secure Socket Layer protocol used by google for secure communication between the end client web browser and provider's web server) [5], [6] and [20]. Due to the failure of the SSL certificate validation, it left the user's Gmail credentials open and eventually some information theft occurred by external adversary. If the Google app engine servers fail to provide the secure communication from the hosted virtual machine (the VM hosting the IoT app to communicate with the SmartFridge from Samsung) to the household device then a "Man in the Middle" attack or "Timing Attack" could take place by an external adversary. Now to compromise the device physically the adversary has to connect with the fridge with a USB port connection to induce a Trojan horse virus or a worm (malicious code).

Even a WiFi enable tracking point Snipper Riffle was breached by a hacker couple Runa and Michael. They demonstrated the vulnerability of that riffle for WIRED in December 2014. A thermal monitoring device from Nest was breached and a ransom was demanded to an urban family in California.

Compromised IoT units could cause eavesdropping, data stealing, or ransomware (most popular these days – hackers compromise

the account and do not release the account until a ransom has been paid). Also a DoS/DDoS could be caused by compromised IoT devices and objects.

3.5 Personal Fitness Apps

One of the biggest applications of cloud computing and the IoT is in the area of personal fitness tracking through smart phones and smart watches [17], and [24]. The three biggest smart device operating systems –Apple iOS, Google Android and Microsoft Phone – have their own apps connecting to a central cloud network to track a user's daily fitness goals and progress update. This service requires a user account in the device, but if the app on the phone and watch are identical, these devices can be synced automatically if either one had a pre-existing account. Here the vulnerable point is if the smart watch is stolen or lost, there lies a possibility of consumer's data retrieval even if the data is residing within the phone.

However, even more prevalent than the smartphone apps are the ever popular personal fitness bands, like the Microsoft Band and the main line of Fitbit products, which are not classified as smart watches only because they can't do much more than monitor what a person's physical actions are and tell time on their screens. But in 2016 January 8th it's been reported by CNBC that Fitbit bands been compromised through the username and passwords. Also the geo location of the consumer has been tracked by external adversaries as well. The customer's activities were traced as well.

The way these smart devices functions are either seeking an internet connection through the smart phone they're paired with, or an open Wi-Fi signal directly, to transmit updated data to the cloud from anywhere there's a signal. So another vulnerability is in the middleware through the connection from the cloud to the device. If the virtual machines in the cloud hosting the device iOS gets compromised then providers of the cloud should take the responsibility.

3.6 Smart Toys

Another place where IoT devices linked to a cloud network are inside children's playthings. These smart toys so far come in the form of intelligent dolls like "My Friend Cayla" and the more prominently known "Hello Barbie" that featured with the ability to response to the child speaking to the toys [6], [10], [13] and [19]. This was accomplished much like how Siri works: the toy receives verbal input from a child which is then stored in a cloud network hub, compiling individual profiles for each child and using these profiles to determine appropriate responses.

Lot of security risks have been presented by these cloud networked toys by security researchers [23]. ToyTalk, the manufacturing company of Hello Barbie claimed the toy to be impenetrable in terms of security breach. Because the way the toy transmits and receives data is similar to SSL encryption used by Google Gmail services, it is unlikely anybody will be able to hack these data requests and replies. Later this claim was proved to be wrong. Along with simultaneously highlighting a key security issue with potentially sensitive data being sent over emails as SSL encryption has been discovered by people working for Google to be exploited through the POODLE (Padding on Oracle Downgraded Legacy Encryption) attack [5]. On October 14, 2014, Google released the details on the POODLE attack, a padding oracle attack that targets CBC – mode ciphers in SSLv3. The vulnerability allows an active MITM attacker to decrypt content transferred on SSLv3 connection. In plaintext, POODLE attack allows a network adversary to extract plaintext of targeted parts of an SSL connection, usually cookie data. In other words, similar to a man-in-the-middle attack (between the web browser and web server) compiling a JavaScript program that can decrypt and extract HTTP information from SSLv3 encrypted communication. For "Hello Barbie", the POODLE attack disclosed not only the authentication mechanism of the doll was weak but also the servers used to communicate with the doll were vulnerable to POODLE, broke the HTTPS encryption.

The most alarming issue is, the toy is technically autonomous and possess the ability to turn on and off while being charged or in active communication with the server for the sake of programming updates. If the HTTPS protocol is breached then it is very easy for an external adversary to gain access to the home network and retrieve information from other IoT objects or devices connected to the home network. Not only data retrieval but also eavesdropping through the security camera is possible. This kind of attack could cause a series of damages on privacy and security.

Another risk of these smart toys is how the data from the doll is managed. The legal agreement for the toy dictates that all voice data received by the toy will be stored in the cloud database for a minimum of two years. This is to allow the algorithm a greater sampling of information to build each individual profile. Of course, the toy's functionality is manageable by a parental supervision account online which handles several of the toy's response settings. Along with parental privilege to view everything recorded by the toy, parents can also delete the recordings. But since the disclaimer from ToyTalk has a loophole in the Service Level Agreement that is access of data to a "Trusted 3rd Party" raised some concerns among parents. They are worried about the possibility of anyone having the access to listen to the recordings of their daughters' dolls if they have access to the server.

The presence of the IoT as an automated service, based on cloud networking is already very evident despite its limited use in consumer products. The benefits of having so many devices connected to the internet will ensure that this technology is only further developed and implemented, expanding the Internet of Things. But the growth rate of the IoT is in some ways a double-edged sword as the technologies to ensure stable security is not increasing proportionally.

4. PHYSICAL AND NETWORK ASPECTS OF VULNERABILITIES IN IOT

No matter how secure a network is, there will always be vulnerabilities as long as two or more devices communicate wirelessly. These vulnerabilities tend to be more widespread across all networks with the existence of IoT objects and devices because the number of individual connections is rising. Also, the increment of heterogeneous networks to provide continuous online services for IoT objects plays a vital role for security vulnerabilities. The complexity of merging network protocols among different network systems often creates security loopholes. These vulnerabilities include both physical and network aspects.

4.1 Physical Aspects

4.1.1 Knockoff Smart Devices

Smart Device knockoffs from suspicious manufacturers can be considered for cloning. The cloning of a smart device lacks of

many functionalities from the original one. For an instance, Apple watch was cloned by a company named Hyperdon with a name A8 in the market [3], [6] and [8]. From pricing stand point A8 is very lucrative since the price is $320 less than an Apple watch. Alarming issue is, A8 works with a jailbroken iPhone. If a malicious code is injected within an A8 then there is possibility of disclosing confidential data (images, videos etc.) from the jailbroken iPhone. Another example is the Samsung UC-UV360 which is compatible with iOS and Android, whereas Moto 360^0 is not, which in terms of security is a good standpoint. But the whole point of using a smart watch collapses if the consumer is unable to utilize the smart features like checking emails, text messages, tracking fitness activities etc. [22] and [21]. Also UC-SBS8 has the ability to insert a SIM card to make phone calls whereas the clone Gear 2 Neo from Korea does not possess this additional feature [23].

Users should be aware of suspicious manufacturers because they can change functionality's of the device to eavesdrop, steal, and monitor the user's activity. Here are some guidance to separate a cloned device from an original one: Branding, Physical Buttons, Screen, Color, Weight, OS, Software version, Battery, Camera Resolution and last but not the least is Price.

4.1.2 Compromised Hardware Installation or Repair

Installation of malicious hardware during installation/repair is a major security issue for IoT objects or devices. In 2006, Japan McDonald launched a promotion with a cup of Coca-Cola soft drink. Every purchase of a drink there was a code. The code was choosing 10,000 lucky winners for an MP3 player loaded with 10 songs. But the marketers failed to anticipate that a QQPass malware was embedded inside the MP3 player. As soon as the device was plugged into the consumer computers, it started to log every keystroke, collecting passwords and gathering information for later transmission [17]. We can imagine the impact if one of the smart devices at a home network is preinstalled with a malware.

4.1.3 Firmware Replacement Attack

Firmware replacement attack can be used when an attacker will replace the firmware update from the user and install a malicious piece of software for the user to download. One of the most recent discovery on spying networks by Equation Group is its mysterious module designed to reprogram or reflash a computer hard drive's firmware with malicious code. According to Kaspersky researchers, it has the ability to subvert hard drive firmware – "Surpasses anything else" they had ever seen. It can also create an invisible storage space to hide data stolen from the system so the attackers can retrieve it later. Kaspersky has so far discovered 500 victims of the Equation Group [11].

4.1.4 Direct Physical Information Extraction

Extraction of security credentials by physically accessing devices can cause an entire network to be compromised. Usually this kind of security breach is committed by a former employee or an employee with ill intension. For an IoT device served by a public cloud could be vulnerable from such kind of security vulnerability from the service provider's perspective. Then only a strong and transparent SLA can end the dispute between the consumer and service provider in a legitimate manner.

4.2 Network Aspects

4.2.1 Eavesdropping

Eavesdropping could take place if the adversary get a hand on the secure key used for data encryption over the network. Now identification of the network protocol or VPN connection type could provide the adversary to conduct an attack with adequate assumption to breach the network security. If an encryption is used then based on the protocol type the encryption algorithm could be guessed wisely. Such as an IPsec VPN connection use Diffie-Hellman encryption usually or a GRE channel VPN connectivity uses DES-3 or AES for encryption. These algorithms are open source techniques and any smart hacker can make the best utilization of the algorithmic knowledge.

4.2.2 Man-in-the-middle Attack

Man-in-the-middle attack takes place during the transmission of data from a source to destination. If an adversary somehow retrieves the shared key between two end points then using the key to generate a bogus data is less complicated. Then broadcasting that data would engage the server to check the authenticity of that packet. A well calculated Timing Attack could reveal network parameters to reveal the network topology.

4.2.3 Routing Attacks

Routing attacks are committed by a network adversary with keen knowledge about the router being used in the network. Usually the attackers target the external border gateway routers since these routers share information between multiple protocols from different partnered companies. Due to heterogeneity of network protocols among the borders, the security measures in these border routers are not strong enough for the complexity of managing large number of heterogeneous traffic passing through every second. Also if the VPN connections are not secure enough between the ISP and the consumer's home network then the GRE channel established to transmit and receive packets is vulnerable to routing attacks.

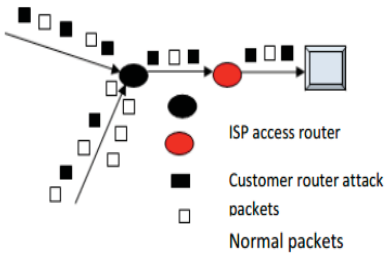


Figure 1: Routing Attack through ISP router

4.2.4 Distributed DoS Attacks

Generally an attacker to conduct Denial of Service or Distributed Denial of Service (DoS/DDoS) attack carts the mission to consume the resources (network bandwidth, throughput etc.) of the system. In DoS/DDoS attack eavesdropping, data stealing, or malware injection are absent. First step of an adversary to commit such type of attack is to obtain an unused legitimate IP address from the IP address pool of the target network. In second step, masquerading himself behind the legitimate source IP address and sends a large number of network packets (TCP, UDP, or http etc.) to the network server for processing. The network server tries to check the authenticity of each packet but actually engaging network resources to verify something vague. As a result

legitimate traffic starve to receive services from the server. Such phenomenon is known as Denial of Service (DoS). If similar attack is committed in a distributed network (Grid, Cloud, Big Data etc.) then due to the load balancing and elastic nature a Distributed DoS attack will take place in the system. A DoS attack could interrupt the routine tasks or even completely shut down the system by consuming all the resources. Such an attack in IoT objects or devices could jeopardize the complete home network and seizing the network accessibility of all the IoT devices connected to that network.

5. ENGAGEMENT OF CLOUD IN IOT

Advances in networking, bandwidth, resource management and virtualization technologies have resulted in service models that involve provisioning computing-as-a-service. Cloud computing, involves cloud service providers (CSP) who offer the services, provisioning and managing a set of technical resources, among tenants: those consuming the cloud services [7], [8] and [10]. The providers' business model is generally to leverage economies of scale by sharing resources between tenants, while tenants gain from being able to pay only for the resources they require, thus removing a costly start-up base and being able to acquire service elasticity—to rapidly scale up and/or scale down resources in response to fluctuations in demand—and more generally, improving access to storage and computational services. The end-user of a system may interact with a cloud provider either directly or indirectly via tenant provided services.

Early works on sensor and communication for IoT devices often mentioned to offload computing and data storage onto a 'server'. With the advancement of distributed technologies, 'server' is replaced with 'cloud'. With cloud comes scalability, high-end virtualization, and elasticity. Many IoT devices are tightly integrated with cloud services. According to a survey, among 38 IoT platforms, 33 relied on cloud or other centralized services [21].

There are many reasons for cloud dependency to support IoT. Some general aspects are mentioned below:

1. Cloud services are 'always on', and with global accessibility, to provide instant location of a device. An example will be the prompt of location update when a carrier has changed the location and consumer is seeking to search something in that new place. In terms of Apple user, iPhone seeks permission for location update as soon as the iCloud needs backup for new storage files etc.

2. As mentioned early, the scalable nature of cloud, makes the synchronization of new IoT objects or devices very efficiently with different data rates and with less technological stress. Since once the IoT platform is setup in cloud then a new device with identical platform will not require much of actions from consumer side. As an example, a consumer can append an iPhone, iMac, iPad, iPod and then later sync an Apple TV or Apple Watch.

3. Load-balancing is another prime feature of cloud, which IoT can take the advantage of. Consumption of resources, like bandwidth, Tx-Rx rates, throughput, storage capacity, battery power etc. Hosting IoT devices in cloud also provides the privilege to scale up or scale down the resources being consumed by the devices depending on the usage. Thus not only achieving an economic service but also deployment of this feature can ensure security by monitoring the usage and running a comparison with the user profile constantly.

In December 2014 an article published by ZDNet, mentioned the security researchers at Malwarebytes Labs, the market leader for anti-malware solution, predicted about the vulnerabilities in cloud operated IoT with malware, ransomware, mobile threats etc.

One of those vulnerabilities is fileless payload. To prevent circumvent detection as well as with more complex obfuscation new types of malware has been created. This malware will not leave a physical file in the storage system but compiles in memory. It is not only difficult to detect and especially challenging for removal of such files.

Another kind is the mobile ransomware. We are familiar with the FBI Moneypak virus which freezes the PC monitor. Already the existence of malware variants are in the market. Encrypting the mobile data and demanding a ransom for data retrieval. Some may argue that a pre-existing backup might suffice. But the consequences will be worse if the backup is breached as well.

Angler Exploit Kit is the first EK to introduce file less exploits and also one of the first one to utilize 0-Day Flash Exploits. Prediction was in 2015, rather than observing frequency of using this kit, it will likely be the main kit to exploit. As an example, for the third time in last two weeks, Adobe has issued an emergency security update for its Flash Player software to fix a dangerous 0-Day threat that adversaries are already exploiting to launch drive-by download attacks [9].

Phishing attacks will get more sophisticated and effective to make users to handover their information. In the most recent campaign of Pawn Storm, several ministers of Foreign Affairs received spear phishing emails. Though the emails contain information about recent events, but in actual fact, these URLs hosted the exploit. Some of the subjects of the emails were; "*Suicide car bomb targets NATO troop convoy kabul*", "*Syrian troops make gains as Putin defends air strikes*", "*Israel launches airstrikes on targets on targets in Gaza*", "*US military reports 75 US-trained rebels return Syria*" etc. [21], [22] and [26].

People are more used to with mobile banking these days. Hence becoming more popular for malware authors to exploit by creating a fake site that looks like original mobile site for banking.

Vigilance is going to be safeguarded by all means, covering everything from cloud to individual mobile devices. Greater vigilance is needed over BYOD any IoT devices that exist within corporations. Also, the idea of breaching a cloud will continue should make companies sit up and pay attention, given the huge fallout from the Sony attack [6], [19], and [21].

6. CONCLUSION

Ubiquity of IoT devices may lead to a translucent society through unified supervision of employees and customers. It represents the next extensive phase of Internet use that will ominously sway consumer comportment and bring about a variety of new offerings from providers to the consumers. IoT is the emerging key technology that paves the way for the next generation of industrial production systems. Smart factories will comprise of self-organizing production systems that optimize themselves with regard to resource availability and consumption, even across company borders.

Though the Internet of Things are underway, yet much of the related scripts are unwritten. Today's IoT systems are not adequately heightened to fulfill the desired functional requirements and to be protected from security and privacy risks. Protection of IoT will require a holistic cyber security framework covering all abstraction layers of heterogeneous IoT systems and across platform precincts. Due to the non-scalable nature of existing security solutions, those are not feasible for large dynamic and scalable network (Cloud, Big Data etc.) of heterogeneous devices and cyber physical systems.

Further research is required to develop and design appropriate IoT security mechanisms, including novel isolation primitives that are resilient to run-time attack.

7. REFERENCES

[1] Ankali, Sanjay B., and D. V. Ashoka, "Detection Architecture of Application Layer DDoS Attack for Internet." N.p., 2011.

[2] Garcia-Morchon, O.; Keoh, S.; Kumar, S.; Hummen, R.; Struik, R. Security Considerations in the IP-based Internet of Things; *IETF Internet Draft draft-garcia-core-security-04*; The Internet Engineering Task Force (IETF): Fremont, CA, USA, 2012.

[3] Kuffner, J.J., 2010, November. Cloud-enabled robots. In *IEEE-RAS International Conference on Humanoid Robotics, Nashville, TN*.

[4] Liu, B., Chen, Y., Blasch, E., Pham, K., Shen, D. and Chen, G., 2014. A holistic cloud-enabled robotics system for real-time video tracking application. In *Future Information Technology* (pp. 455-468). Springer Berlin Heidelberg.

[5] Möller, B., Duong, T. and Kotowicz, K., 2014. This POODLE bites: exploiting the SSL 3.0 fallback. *Google, Sep*.

[6] Suresh Kumar, S., Gollakota, S. and Katabi, D., 2012. A cloud-assisted design for autonomous driving.

[7] Jordan, S., Haidegger, T., Kovács, L., Felde, I. and Rudas, I., 2013, July. The rising prospects of cloud robotic applications. In *Computational Cybernetics (ICCC), 2013 IEEE 9th International Conference on* (pp. 327-332). IEEE.

[8] Sadeghi, A.R., Wachsmann, C. and Waidner, M., 2015, June. Security and privacy challenges in industrial internet of things. In *Proceedings of the 52nd Annual Design Automation Conference* (p. 54). ACM.

[9] Weinberg, B.D., Milne, G.R., Andonova, Y.G. and Hajjat, F.M., 2015. Internet of Things: Convenience vs. privacy and secrecy. *Business Horizons*, *58*(6), pp.615-624.

[10] Díaz, M., Martín, C. and Rubio, B., 2016. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications*.

[11] Sicari, S., Rizzardi, A., Grieco, L.A. and Coen-Porisini, A., 2015. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, pp.146-164.

[12] Granjal, J., Monteiro, E. and Sa Silva, J., 2015. Security for the internet of things: a survey of existing protocols and open research issues. *Communications Surveys & Tutorials, IEEE*, *17*(3), pp.1294-1312.

[13] Weber, R.H., 2015. Internet of things: Privacy issues revisited. *Computer Law & Security Review*, *31*(5), pp.618-627.

[14] Weber, R.H., 2010. Internet of Things–New security and privacy challenges. *Computer Law & Security Review*, *26*(1), pp.23-30.

[15] Baldini, G., Botterman, M., Neisse, R. and Tallacchini, M., 2016. Ethical Design in the Internet of Things. *Science and Engineering Ethics*, pp.1-21.

[16] Botta, A., de Donato, W., Persico, V. and Pescapé, A., 2016. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, *56*, pp.684-700.

[17] Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), pp.431-440.

[18] O'Neill, M., 2014. The Internet of Things: do more devices mean more risks? *Computer Fraud & Security*, *2014*(1), pp.16-17.

[19] Kang, R., Dabbish, L., Fruchter, N. and Kiesler, S., 2015. "My Data Just Goes Everywhere:" User Mental Models of the Internet and Implications for Privacy and Security. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)* (pp. 39-52).

[20] Nguyen, K.T., Laurent, M. and Oualha, N., 2015. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks*, *32*, pp.17-31.

[21] Singh, J., Pasquier, T., Bacon, J., Ko, H. and Eyers, D., 2015. Twenty security considerations for cloud-supported Internet of Things.

[22] Wen, Q., Dong, X. and Zhang, R., 2012, October. Application of dynamic variable cipher security certificate in internet of things. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on* (Vol. 3, pp. 1062-1066). IEEE.

[23] Suo, H., Wan, J., Zou, C. and Liu, J., 2012, March. Security in the internet of things: a review. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on* (Vol. 3, pp. 648-651). IEEE.

[24] Altolini, D., Lakkundi, V., Bui, N., Tapparello, C. and Rossi, M., 2013, July. Low power link layer security for iot: Implementation and performance analysis. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International* (pp. 919-925). IEEE.

eGovernment service security policy: obligation conflict resolution in XACMLv3

Ibrahim Yonis Omar, Romain Laborde, Ahmad Samer Wazan, François Barrère, Abdelmalek Benzekri
Institut de Recherche en Informatique de Toulouse
University Paul Sabatier
Toulouse, France
{Yonis, Romain.Laborde, Ahmad-Samer.Wazan, Francois.Barrere, Abdelmalek.Benzekri}@irit.fr

Abstract— Today, many governments tend to propose e-services to their citizens. However, implementing an eGovernment environment shall face up to several security challenges including integrating security requirements coming from multiple stakeholders. In this article, we analyze the conflicts that can occur between eGovernment security requirements. Since these security requirements can contain both authorizations and obligations, we cover these two aspects. Then, we propose a new conflict resolution algorithm that handles conflicts between authorizations as well as obligations. This work has been implemented in XACMLv3.

***Keywords*— eGovernment; Access control; Obligations; conflict; XACMLv3**

I. INTRODUCTION

Towards a reduction process of gaps between user expectations and public services, public administration tends to use ICT in order to offer efficient services. This paradigm is known as electronic Government (eGovernment) [1] and can be classified according to different target areas as:

- Government-to-Government (G2G), also known as e-administration, refers to electronic collaboration between different government agencies,
- Government-to-Citizen (G2C), is the process that electronically provides on-demand and personalized public services to citizens,
- Government-to-Business (G2B), sets up online relationship between government and the business sector in order to interactively provide information on regulations, advice, and procedures.

This modernization of relations with a government takes an interest because it is generally offered in a centralized way – with a one-shop portal: all eGovernment services are available in one place and exposed from a common portal [2].

eGovernment services are classified depending on its levels of paperless known as maturity level. Designing models of maturity levels has been the subject of several studies [3]. Although the number of phases differs from one model to another, all the models are based on four main phases to measure the maturity of a system of e-government. These main level starts from level 1, with a simple informational website to level 4 with an advanced shared services between public administration.

However, implementing eGovernment must address several challenges [4]. Among them the way to design and write a security policy remains complex [5] because it must consider different High Level Security Requirements (HLSRs) given by stakeholders [6]. Security policy of an eGovernment service must comply at the same time with HLSRs expressed by law issuers (Li), Executive governance (Eg) and government departments (Gd).

Our proposed research is done in the context of the Djiboutian eGovernment Cloud Community (eGCC) which aim to implement a G2G infrastructure: two main issues have been highlighted when the common security policy of eGCC was analyzed. .

First, according to its own area of occupation, each stakeholder expresses its HLSR using an expression model that may differ from other stakeholders. DAC (Discretionary Access Control) - – MAC (Mandatory Access Control) model; - RBAC (Role Based Access Control) [7] [8][9] are some example of such models. While specific constraints must be considered, it remains that the policy within eGCC must adopt unified way of expression.

This issue was discussed in [10] where we proposed the usage of ABAC (attributes based access control) with XACMLv3 [11] standard. A common policy-based language for eGovernment was presented in order to express multiple specific constraints (thanks to ABAC) and applied our approach to an open source Cloud Computing solution – OpenStack [12].

The second issue is related to the consistency of HLSRs. HLSRs can contain both authorizations (*permission on resources with defined conditions*) and obligations (*duties to execute*). The common security policy for eGovernment service is established by combining all stakeholders' HLSRs. Each HLSR, written in XACMLv3 is delivered by stakeholders and contain specific constraints. As a consequence a simple HLSR combining may result into conflicts and inconsistencies.

Many works have studied conflicts between authorizations. E.g., XACMLv3 provides twelve authorization conflict resolution algorithms. However, much less researches have explored conflicts between obligations and how to manage these conflicts. As consequence, XACMLv3 doesn't include any obligation conflict resolution. In this article, we analyze the obligation conflict management issue in the context of

eGovernment that involves multiple stakeholders. Also, we propose an obligation conflict resolution algorithm that we implemented in XACMLv3.

The rest of the article is structured as follows. In section II, we present security management issues in eGovernment. In section III, we present XACMLv3 and its capability to express eGovernment policy security requirements. In section IV, we introduce our approach to enhance XACMLv3 with an algorithm for eGovernment obligation conflict management. In section V, we list some related work. Finally, we draw our conclusion and perspectives in section VI.

II. EGOVERNMENT SECURITY MANAGEMENT

The security of eGovernment services is governed by a set of HLSRs that we classify according to the institution source: legal, governance and business.

Legal HLSRs are expressed from legislation and concerns compliance to legal texts applied to information and data collected by public administration. Data sensitivity in the context of eGovernment requires regulation. To prevent abuse of data usage in administrative procedures and thus establish trust between the users and the e-Government service, a number of laws have been voted and must be respected.

Governance HLSRs is expressed from executives and ensure the proper organization of security within eGovernment organizations IS. It corresponds to the general policy of government on eGovernment and expresses requirements on how eGovernment is implemented.

Business HLSRs expressed from organization are essentially dealing with business needs. They are driven by the profession's needs of ministerial departments.

Given the multiple policies with its HLSR, security compliance of eGovernment services to those policies may be subject to conflicts. We propose to address these conflicts by prioritizing them according to their sources. This priority is based on the natural hierarchy characterizing the machinery of government

Legal HLSRs should have greater weight than those dictated by the executives and business stakeholders. Executive HLSRs should have more weight than those of business.

A weight is given to each HLSR according to its source (i.e., legal, governance or business). Based on that weight, HLSRs are prioritized to resolve the conflicts. Thus, the order relation that states is:

Legal Policies (LP) > Executive Policies (EP) > Business Policies (BP).

To highlight HLSRs consistency challenge, let us consider, for instance, the Tax Income Public Agency (TIPA) that provides tax information and services. In order to enhance its service treatments, TIPA decide to offer an eGovernment services. At the first stage of this migration, TIPA will only provide informational eGovernment service (Maturity level 1). Thus TIPA creates Virtual Machines (VM) which hosts a web server within a virtual data center (VDC), offered by an eGovernment Cloud provider (eGCp).

Resources of TIPA is governed by a set of policies with multiple HLSRs from 1) Law, 2) Executive and 3) Business. Enforcement of policies must follow the order of the predefined hierarchy: Legal (LP)> Executives (EP) > Business (BP). Let's consider that the policy of TIPA is the following:

- LP1: Identifiable data collected from users shall not be transferred or used in any other purpose without the prior consent of its user.
- LP2: Regulation *requires* encrypting eGovernment services resources.
- BP1: Resource encryption is required for eGovernment service classified as Maturity Level 3 [3] and above only.
- BP2: In case of cyber attack any executive administrative task of eGovernment system should not be available except for Chief Information Security Officer (CISO).
- EP1: In case of cyber attacks to eGovernment system, access to the system is forbidden until constitution of a team by ministerial Decree.

Clearly, the policy of TIPA entails the handling of different HLSRs coming from different sources and the preservation of authorizations and obligations orders. Thus, the following criteria must be filled:

Criterion 1 — There must have policies hierarchy management systems.

Criterion 2 — Security management system has to apply both authorizations and obligations policies.

Criterion 3 — Regardless of policy selection order, enforcement must respect the predefined hierarchy.

In order to handle policies whose expressions (e.g., RBAC, MAC or DAC) and sources (Law, Executives and Business) are different, we have selected the language XACMLv3 to implement our solution. The extensibility of this language permits us to enhance its capability to the eGovernment context.

III. XACMLV3 AND EGOVERNMENT POLICY SECURITY HLSR

We briefly present in this section the XACMLv3 standard and how it could meet the constraint of HLSRs.

XACML (eXtensible Access Control Markup Language) version 3 is an XML-based specification for access control that has been standardized by OASIS [8]. XACMLv3 describes an architecture, an attribute-based access control policy language and a request/response language.

The XACMLv3 policy language is used to describe general access control constraints in terms of constraints on attributes. Specifically, attributes could be any characteristics of any category such as the subject, the resource, the action, or the

environment in which the access request is made. Attributes have an identifier, which is a Uniform Resource Name (URN), and a data type also identified by a URN. Considering attributes makes the language very flexible. Moreover, XACMLv3 language is natively extensible. A XACMLv3 policy is composed of:

- A *target* element which is a first filter for searching the applicable policy
- A set of *obligation expressions* that are instantiated when a matching request is processed. PEPs must enforce obligations.
- A set of *advice expressions* that are instantiated when a matching request is processed. Advice is similar in its form to an obligation. However, PEPs may or may not enforce advice.
- A set of *rules* that are expressions to determine if a request is denied or permitted. A *rule* contains a *target* and may include *obligations* and *advice* specific to this rule.
- Policies can be grouped in *policy sets*.

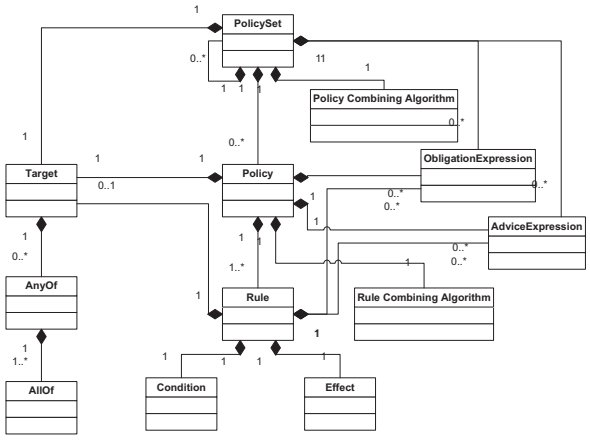


Figure 1. The XACMLv3 policy language mode [11]

The architecture of XACMLv3 consists mainly in two management components: the Policy Enforcement Point (PEP) and the Policy Decision Point (PDP). The PEP is the guard of the resources. It intercepts the request expressed in the native application format, translates it into the XACMLv3 request format and sends it to the PDP. The PDP is the "brain" of the system. Once it receives a request from a PEP, it looks at its XACMLv3 policy for matching rules. Each rule leads to a specific decision, which is a triplet (permit/deny, set of obligations, set of advice). If only one rule match then the decision is applied .If the request matches two or more rules, the PDP builds a unique decision by applying the rule combining algorithms and the policy combining algorithms. This unique decision is then returned to the PEP that enforces it in the actual system.

XACMLv3 includes a set of predefined *policy/rule combining algorithm*s, used to resolve the eventual conflicts in the authorizations:

- Deny overrides: This algorithm combines decisions of policies / rules so that if any decision is *Deny*, then that decision is applied.
- Permit-overrides: This algorithm does the same work as the above algorithm, but in this case *Permit* decisions are the dominant ones.
- First applicable: This algorithm applies the first decision (Deny or Permit) found and returns the first match as result.
- Only one applicable: This algorithm is used only for combining policies. It cannot be used to combine rules.
- Deny unless permit: The algorithm result will be Deny unless an explicit Permit Decision is found.
- Permit unless Deny: Same as the above algorithm except default result will be Permit unless explicit Deny is found.

These algorithms also exist in *ordered* mode where policy, *policy set* and *rules* are considered in the order in which they are defined. Thus, prior establishment of hierarchy can be fulfilled with XACMLv3 predefined combining algorithms.

Although XACMLv3 supports natively authorization conflict management with its combining algorithms, these algorithms don't take into account the obligations. In the scenario proposed above, as VM creation is authorized for TIPA, BP1 and LP2 HLSRs obligations conflict. LP2 (law) requires all VMs must be encrypted on creation. BP1 (business) does not claim such encryption as the service provided is informational (maturity level1) and is not an advanced one (maturity level3). As XACMLv3 does not have an obligation conflict management algorithm, such obligations are together sent to PEP which generates a problem of applicability for PEP or obligation Service unless formal handling methodology.

IV. A NEW CONFLICT RESOLUTION ALGORITHM THAT CONSIDERS OBLIGATIONS

In this section we present our new algorithm to resolve eGovernment obligations conflict issues in XACMLv3. A conflict resolution algorithm for obligations consists of two parts: conflict detection and conflict resolution.

A. Detection of Conflict

We represent a XACML rule $R \in RULES$ as a triple $(cond_R, effect_R, obligations_R)$ where $cond_R \in COND$ is a Boolean expression with free variables, $effect_R \in$ {Permit, Deny} and $obligations_R \in \mathbb{P}$(OBLIGATIONS) is a set of obligation expressions with free variables.

Evaluating a rule R for a given request *req* can be achieved by executing three tasks:

1. The bounding of the free variables of the condition using the request attributes. We note the bounded condition with bound($cond_R$,req).
2. The interpretation of the bounded condition that provides a Boolean value (the condition matches or not). We represent it by interpret (bound($cond_R$,req)).
3. The bounding of the free variables of the obligations using the request attributes. We note the bounded obligations with bound($obligation_R$,req).

Detecting a conflict for a given request *req* can then be formalized as follows:

$\exists$ (R1, R2) $\in$ RULES^2,

R1= *($cond_{R1}$, $effect_{R1}$, $obligations_{R1}$), R2 = ($cond_{R2}$, $effect_{R2}$, $obligations_{R2}$)* ,

interpret(bound($cond_{R1}$,req)) = $\top$ ⅄

interpret(bound($cond_{R2}$,req)) = $\top$

where at least one of the following two conditions is true:

Condition 1) $effect_{R1} \neq effect_{R2}$

Condition 2) $\exists$ (o_{R1}, o_{R2}) $\in$ $obligations_{R1} \times obligations_{R2}$,
conflict(interpret(bound(o_{R1},req),
interpret(bound(o_{R2},req)))

Detecting a conflict must be performed at the decision stage, i.e. by the PDP, in order to provide a unique decision to the PEP. When rules don't include obligations, the detection is easy since the PDP natively performs interpret(bound($cond_R$,req)) and evaluating *condition 1* requires only to compare two values (Permit/Deny).

However, evaluating *condition 2* is not as simple as *condition 1*. In fact, obligations conflict detection requires the analysis of the semantic of the obligations and a dynamic detection of possible conflict is not obvious. E.g., what is the result of interpret (bound(o_{R1},req))? How to detect the execution of an obligation is conflicting with another one? Such issue is pointed with BP1 against LP2. Thus, for obligation conflict detection, we use the follow manual discovery algorithm to handle semantic means of obligation action.

Algorithms 1 ObligationConflictsDetect()

1 Let p *be the parameter of an eGovernment service resource* Obli *represent, for each i, 1≤ i ≤, a set of obligations applied to* p *and* P_{Obl} *possible obligation conflict.*

2 P_{Obl} = *if* $\exists$(OblBP,OblLP) x p(TIPA)

3 if $\exists$ P_{Obli} ⁄\ OblBP ¬ OblLP ➔ OblC

End Algorithms

Our algorithms detect obligation conflict, if two or more obligations are designed towards the same parameters of single resource to the same eGovernment services (TIPA). For TIPA, for instance, we identify the set of obligation HLSR applied to it. Afterwards, we identify whether any of the obligations potentially conflict with each other. If positive conflict match, we select conflicting obligations.

B. Obligation conflict resolution

To detect and resolve obligation conflicts in XACMLv3, we propose to extend the PDP with obligation inconsistency management algorithms. We adopt answer set programming (ASP), a form of declarative programming [13], to formally represent our model. ASP is based on the stable model semantics of propositional logic programming and allows non-monotonic reasoning. Syntactically, ASP is closed to Prolog. However, instead of asking a question and using inference to find the solution like in Prolog, ASP grounds the variables and computes stable models (for more details [14]).

We recall quickly some basics on the ASP syntax. Rules are of the form "`h :- b.`" where `h` is the head and `b` is the body. It can be understood as if predicate `b` is true in an answer then `h` is also true in the answer. When the rule has no body, for example "`h.`" then `h` is a fact and must be in all the answers. When the rule has no head, for example "`:- b.`", the rule is a constraint and means that `b` must not be true in any answers. Finally, it is also possible to specify choice. For example, the following rule "`{h1;h2}:-b.`" can be understood as if `b` is true then there can be an answer where `h1` is true and another one where `h2` is true.

```
#show finalDecision/1.
%issuer/1 is the source of the obligation.
issuer(law).
issuer(executive).
issuer(business).
%superior/2 defines that LAW > EXECUTIVE > BUSINESS.
superior(issuer(law), issuer(executive)).
superior(issuer(executive), issuer(business)).

%%%%%%%%%%%
% GUESS
%%%%%%%%%%%

%conflict/2 defines explicitly conflicts between two obligations. This relation is a symetric.
conflict(obligation(OBL1), obligation(OBL2)) :- conflict(obligation(OBL2), obligation(OBL1)).

% candidate solution : Any obligation can be in the final decision
{finalDecision(obligation(OBL)): decision(obligation(OBL), issuer(_))}.

% If 2 obligations are in conflict then choose the one from the requirement with higher priority
finalDecision(obligation(OBL1)) :- conflict(obligation(OBL1), obligation(OBL2)), decision(obligation(OBL1),issuer(F1)), decision(obligation(OBL2),issuer(F2)), superior(issuer(F1), issuer(F2)).
% Otherwise choose one
1{finalDecision(obligation(OBL1));finalDecision(obligation(OBL2))} :- conflict(obligation(OBL1), obligation(OBL2)), decision(obligation(OBL1),_), decision(obligation(OBL2),_).

%An obligation that has no depency issue can be in the final decision
finalDecision(obligation(OBL)) :- not conflict(obligation(OBL), obligation(_)), decision(obligation(OBL),_), not dependencyIssue(obligation(OBL)).

%%%%%%%%%%%
% CHECK
%%%%%%%%%%%

%Two conflicting obligations cannot be in the final decision
:- conflict(obligation(OBL1), obligation(OBL2)), finalDecision(obligation(OBL1)), finalDecision(obligation(OBL2)).

%There is a depency issue when an obligation depends on another one that is not in the final decision
dependencyIssue(obligation(OBL1)) :- dependsOn(obligation(OBL1),obligation(OBL2)), not finalDecision(obligation(OBL2)).

%An obligation that has some depency issue cannot be in the final decision
:- finalDecision(obligation(OBL)), dependencyIssue(obligation(OBL)).
```

Figure 2. Our Answer Set Program for resolving conflicts.

Since determining matching XACMLv3 rules and detecting authorization effects conflict is already done by any XACMLv3 PDP, we focus only on obligation conflict resolution. Thus, we consider that a set of rules matches a specific XACMLv3 request and these rules have the same effects. However, some rules contain obligations.

For our implementation, we used clingo 4 [15]. We followed the guess and check methodology [14] which consists in:

1) **Guess** : Create candidate solutions to the problem
2) **Check**: Check with rules/constraints whether a candidate solution is valid or not.

Thus, based on a set of obligations as input, we generate candidate solutions (**Figure 2**). If conflicts exist, we choose the obligation with the higher priority calculated based on the issuer (Law > Executive > Business). We then check if there is no conflicting obligations or functional dependency issue in a candidate solution. Conflicting obligations and obligation dependencies have to be manually expressed using predicates *conflict*/2 and *dependsOn*/2 (Figure 3). This means that all obligations applied on eGovernment services must be predetermined and analyzed to produce this data.

```
%%%%%%%%%%
% Example of an initial knowledge database
%%%%%%%%%%
obligation(obl1).
obligation(obl2).
obligation(obl3).
obligation(obl4).
% there is a conflict between obl2 and obl3
conflict(obligation(obl2), obligation(obl3)).

% dependsOn/2 represents the dependency between obligations.
% obl4 depends on obl3
dependsOn(obligation(obl4),obligation(obl3)).
```

Figure 3. Example of an initial knowledge database for conflict resolution.

```
% translated from the matching XACML rules
decision(obligation(obl1), issuer(law)).
decision(obligation(obl2), issuer(law)).
decision(obligation(obl3), issuer(executive)).
decision(obligation(obl4), issuer(executive)).
```

Figure 4. Predicates translated from matching XACMLv3 rules.

Finally, when the PDP has to take a decision, it translates the candidate obligation into predicate *decision*/2. **Figure 4** gives an example where obligations *obl1* and *obl2* are coming from law HLSRs and obligation *obl3* and *obl4* from executive HLSRs. After being processed, the final decision calculated by the ASP program is cancelled *obl3* (in conflict with *obl2* that has higher priority) and *obl4* (it depends on *obl3*).

C. Obligation enforcement planning

We complete the obligation conflict resolution with an obligation enforcement planner to ensure that obligations are executed in the right order (compliance to our criterion3). Indeed, unwanted side effects may arise if obligations are applied in any arbitrary order. We propose to specify known side effects using predicate *before*/2 meaning that an obligation must be applied before another one (Figure 5). Using the methodology Guess&Check, we build the following obligation enforcement planner. For example, if a final decision consists in applying obligations *obl1*, *obl2*, *obl3*, *obl4*, planner proposes several solutions like the following sequences <*obl1*, *obl3*, *obl4*, *obl2*> or <*obl3*, *obl1*, *obl2*, *obl4*>

```
#show enforce/2.
%%%%%%%%%%
% Example of an initial knowledge database
%%%%%%%%%%

% before/2 represents that fact that an obligation must be executed before another one.
before(obligation(obl1),obligation(obl2)).
before(obligation(obl3),obligation(obl2)).
before(obligation(obl5), obligation(obl2)).

%%%%%%%%%%
% GUESS
%%%%%%%%%%

step(X):- X=1..N, N=#count{OBL:obligation(OBL)}.

{enforce(obligation(OBL),step(T)): obligation(OBL), step(T)}.

%%%%%%%%%%
% CHECK
%%%%%%%%%%

% Any obligation of the XACML decision must be enforced.
:- not enforce(obligation(OBL),step(_)), obligation(OBL).

% An obligation that has been defined to be executed before another one cannot be
enforced after.
:- enforce(OBL1,step(T1)), enforce(OBL2, step(T2)), T1 < T2, before(OBL2, OBL1).

% Only one obligation can be enforce at some time T.
:- enforce(obligation(OBL1),step(T)), enforce(obligation(OBL2), step(T)), OBL1 != OBL2.

% An obligation is enforced only once.
:- enforce(obligation(OBL),step(T1)), enforce(obligation(OBL), step(T2)), T1 != T2.
```

Figure 5. Our obligation enforcement planner

V. RELATED WORK

eGovernment security policy.

Security in eGovernment is largely acknowledged as a challenge [16] [17][18] . As part of a European project, Lambrinoudakis et al. [19] propose PKI-based security policy for eGovernment services. According to eGovernment service, its level of paperless and users involved, a risk level, which can be low, medium or high, is labeled. Based on this level, they define security requirements. They then deal with these levels of requirements with a PKI-based security policy.

Drogkaris et al. [20] have acknowledged privacy concerns in eGovernment security policy with user preference involvement. They propose a Privacy Controller Agent (PCA), an engine that manages privacy enforcement in eGovernment. They underline existing of various rules in the service provider privacy policy document. For Drogkaris et al., conflict can occur between service provider and user preferences. Although this approach is dealing with the security concerns (privacy aspect) of modern eGovernment with centralized one stop shop portal, it ignores the potential conflict between various inherited rules expressed by services provider policies.

A. XACMLv3 conflicts analysis

Hwang et al. [21] propose a tool that generates the XACML-represented policy and check the consistency of these policies both statically and automatically. Verification focuses on policy coherence, specifically whether the authorization result is produced as expected or not.

To detect inconsistencies and conflicting XACML-represented policy, Martin and Logrippo [22] use Alloy [23] a first order logic model checking tools. They represent XACML element as a logical model and translated into Alloy in order to

detect inconsistency of policies. Inconsistency is produced when "two rules return two different decisions (permit and deny) in a context of a specific request".

Fisler et al. [24] propose verification and validation policy tool Margrave (ref) for XACML-represented policy. With verifier component integrated into margrave, different possible decisions from XACML policy are represented as a form of diagram and are verified to detect the eventual conflict between decisions.

Mohan et al. [25] highlight the problematic of authorization in taxonomy-based biomedical databases. They propose strategies and algorithms to detect policies conflict and potential inference attacks resulting from how policies are formulated. Their proposition is implemented in XACML.

Martin et Xie [26] determine the gap between result of decision and expected behavior of policies written in XACML by generating request on policies and use the responses as input to a tool using machine learning algorithms. As an output these tool generate behaviors of policies by listing, "inferred properties that may not be true for all requests but are true for most requests in order to highlight possible special case requests.

However, all these works have uniquely focused on the management of inconsistency and conflicts of the authorization side of XACML. We have shown that conflicts in policies can also be produced because of opposed obligations to carry out. Since XACMLv3 takes into account the obligation representation and due to the lack of obligation conflict management in the current works, we have proposed an algorithm to detect and resolve the eventual conflicts produced by different opposed obligations in XACMLv3 policies in the eGovernment context.

VI. CONCLUSION

Managing the security policies in context of eGovernment entails the construction of a security management system that allows to: 1) Combine different policies expressed by different models, 2) Handle conflict decisions produced at policy and rule levels (using combining algorithms), and 3) Handle conflict obligations.

In a previous work [10], we addressed the first point. In the current work, we have handled the second and third points. Specifically, we have exploited the existing capabilities of XACMLv3 to address the second point. However, since XACMLv3 does not support natively obligation conflict management, we propose an obligation conflict management algorithm that can be executed by a PDP. Also, we have implemented an obligation planner intended to preserve obligation orders.

Our contribution didn't consider the real security state of different stakeholders. Indeed, the higher the maturity levels of eGovernment services are, the more resources are available on the Internet. Also, advanced high level eGovernment services require involvement of multiple stakeholders. Thus, the security of these resources becomes an essential matter to consider. However, due to divergence state of security preparation of stakeholders, defining formal security responsibility of stakeholders towards advanced eGovernment service is not obvious. How we can determine security responsibility of involved stakeholders? Thus, defining a scale of security competency levels of stakeholders may help to preserve the security of the advanced high-level eGovernment services. We believe that such levels can be considered as conditions to delimit the scope of each stakeholder. These issues constitute the main activity that we are conducting currently.

VII. REFERENCE

[1] M. J. Moon, "The evolution of e-government among municipalities: rhetoric or reality?," *Public Adm. Rev.*, vol. 62, no. 4, pp. 424–433, 2002.

[2] A. Tat-Kei Ho, "Reinventing local governments and the e-government initiative," *Public Adm. Rev.*, vol. 62, no. 4, pp. 434–444, 2002.

[3] K. Layne and J. Lee, "Developing fully functional E-government: A four stage model," *Gov. Inf. Q.*, vol. 18, no. 2, pp. 122–136, 2001.

[4] D. S. Jones and B. Crowe, *Transformation not automation: The e-government challenge*. Demos, 2001.

[5] M. Al-Sebie and Z. Irani, "Technical and organisational challenges facing transactional e-government systems: an empirical study," *Electron. Gov. Int. J.*, vol. 2, no. 3, pp. 247–276, 2005.

[6] J. Rowley, "eGovernment stakeholders—Who are they and what do they want?," *Int. J. Inf. Manag.*, vol. 31, no. 1, pp. 53–62, 2011.

[7] G.-J. Ahn, "Discretionary Access Control," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Springer US, 2009, pp. 864–866.

[8] P. Samarati and S. D. C. Di Vimercati, "Access control: Policies, models, and mechanisms," *Lect. Notes Comput. Sci.*, pp. 137–196, 2001.

[9] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," *ArXiv Prepr. ArXiv09032171*, 2009.

[10] I. Y. Omar, R. Laborde, A. S. Wazan, F. Barrere, and A. Benzekri, "G-Cloud on Openstack: Adressing access control and regulation requirements," in *International Symposium on*,*Networks, Computers and Communications (ISNCC),* 2015, pp. 1–6

[11] "eXtensible Access Control Markup Language (XACML) Version 3.0." [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html. [Accessed: 08-Feb-2016].

[12] "Documentation — OpenStack." [Online]. Available: https://wiki.openstack.org/wiki/Documentation. [Accessed: 08-Feb-2016].

[13] V. Lifschitz, "What Is Answer Set Programming?.," in *AAAI*, 2008, vol. 8, pp. 1594–1597.

[14] T. Eiter, G. Ianni, and T. Krennwallner, *Answer set programming: A primer*. Springer, 2009.

[15] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Clingo= asp+ control: Extended report," Technical report, University of Potsdam, 2014.

[16] Z. Zhou and C. Hu, "Study on the e-government security risk management," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 5, pp. 208–213, 2008.

[17] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Security requirements engineering for e-government applications: analysis of current frameworks," in *Electronic Government*, Springer, 2004, pp. 66–71.

[18] R. Breu, M. Hafner, B. Weber, and A. Novak, "Model driven security for inter-organizational workflows in e-government," in *E-Government: Towards Electronic Democracy*, Springer, 2005, pp. 122–133.

[19] C. Lambrinoudakis, S. Gritzalis, F. Dridi, and G. Pernul, "Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy," *Comput. Commun.*, vol. 26, no. 16, pp. 1873–1883, 2003.

[20] P. Drogkaris, S. Gritzalis, C. Kalloniatis, and C. Lambrinoudakis, "A Hierarchical Multitier Approach for Privacy Policies in eGovernment Environments," *Future Internet*, vol. 7, no. 4, pp. 500–515, 2015.

[21] J. Hwang, T. Xie, V. Hu, and M. Altunay, "ACPT: A tool for modeling and verifying access control policies," in *Policies for Distributed Systems and Networks (POLICY), 2010 IEEE International Symposium on*, 2010, pp. 40–43.

[22] M. Mankai and L. Logrippo, "Access control policies: Modeling

and validation," in *5th NOTERE Conference (Nouvelles Technologies de la Répartition)*, 2005, pp. 85–91.
[23] D. Jackson, "Alloy 3.0 reference manual," *Softw. Des. Group*, 2004.
[24] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz, "Verification and change-impact analysis of access-control policies," in *Proceedings of the 27th international conference on Software engineering*, 2005, pp. 196–205.
[25] A. Mohan, D. M. Blough, T. Kurc, A. Post, and J. Saltz, "Detection of conflicts and inconsistencies in taxonomy-based authorization policies," in *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*, 2011, pp. 590–594.
[26] E. Martin and T. Xie, "Inferring access-control policy properties via machine learning," in *Policies for Distributed Systems and Networks, 2006. Policy 2006. Seventh IEEE International Workshop on*, 2006, p. 4–pp.

Modeling Host OSI Layers Cyber-Attacks Using System Dynamics

Uma Kannan[1], **Rajendran Swamidurai**[2], **and David Umphress**[1]
[1]Computer Science and Software Engineering, Auburn University, Auburn, AL, USA
[2]Mathematics and Computer Science, Alabama State University, Montgomery, AL, USA

Abstract - *Cyber security modeling is the process of creating a normalized view of the cyber security situation. A typical cyber security model has information about the network infrastructure, security settings, and a list of possible vulnerabilities and threats. By using known vulnerabilities, and information about the infrastructure and security controls in place, the cyber security simulation allows an organization to imitate the attacker activities and helps to assess the system's risk exposure. Networks are normally modeled or simulated through discrete-event techniques. But the discrete-event simulations can only simulate a few seconds worth of network operations and the primary focus of discrete-event models is on packet traffic. This means that cyber-attacks/defenses are viewed from the network layer, layer 3, in the OSI model. This obscures more insidious attacks at higher layers in the OSI model. This paper presents a study which models a computer network as a systems dynamic model to explore more insidious cyber-attacks and the resulting system-level effects that might occur on host OSI layers, layer 4 and above, in the OSI model.*

Keywords: Cyber security; cyber security modeling; system dynamics; continuous simulation; simulation and modeling; cyber-attacks/defenses.

1 Introduction

Modeling is the process of capturing the key characteristics or behavior of a real world system under study and it helps us in understanding the essential parts of a system and the relationship between them [1-3]. "*Cyber security modeling is the process of creating a normalized view of the cyber security situation.*" [4] A typical cyber security model has information about the network infrastructure, security settings, and list possible security vulnerabilities and threats [4]. Simulation is the process of imitating a system, based upon our knowledge or assumptions about the behavior of the parts of a system, in order to get the insight of a whole system [5]. Similarly, by using known vulnerabilities and the current knowledge about infrastructure and security controls, the cyber security simulation allows an organization to imitate the attacker activities and helps to assess the system's risk exposure [4].

Networks are normally modeled or simulated through discrete-event techniques, in which the state of system changes only at discrete points in time. Depending on the granularity of the model, this means simulating the movement of packets throughout a network and measuring such things as throughput, latency, etc. In discrete-event simulation (DES), cyber-attacks are simulated by altering the flow or rate of packets and observing the result.

Discrete-event network simulation tools such as cnet, EcoPredictor, IT SecisionGuru, NetCracker, and NetRule are used by professional system administrators and systems application designers to model and analyze packet traffic, buffer overflow, operating system compromise, and so on. [1]. With respect to information security, these network simulation tools are normally used to model tasks such as server availability and router availability. They also used to make the in depth analysis of authentication server's loads and unusual network traffic [1].

DES approach has two flaws. First, simulations can only simulate a few seconds worth of network operations due to the massive number of packets that are transmitted during normal operations. Second, these models focus primary on packet traffic. This means that cyber-attacks (and the resulting cyber defenses) are viewed from the network layer, that is, layer 3 in the open system interconnection (OSI) model. This obscures more insidious attacks at higher layers in the OSI model.

This paper presents a study which models a computer network as a systems dynamic model (a.k.a. continuous simulation). Its objective is to explore more insidious cyber-attacks and the resulting system-level effects that might occur on host OSI layers (layer 4 and above); that is, on transport, session, presentation, and application layers. For modeling we have used the concept of System Dynamics (SD), because it allows us to see systemic effects – something that is not feasible with DES. In SD methodology, the stock-flow diagram is used depict the underlying mathematical model, the model structure and the interrelationships between its components. Once the underlying mathematical structure is captured, the stock-flow diagram can be easily translated into system of differential equations, and simulated using continuous simulation software such as Powersim.

Section 2 describes the system dynamics methodology. Section 3 explains the benefit of using modeling and simulation technique in cyber security in detail. Section 4 presents the system dynamics cyber security modeling/simulation process. Section 5 shows an example cyber security attack simulation model. Section 6 shows the results. Summary and conclusion is presented in Section 7.

2 System Dynamics

System dynamics (SD) [6] is a methodology used to study a system change over time. In SD, a system is defined as a collection of interacting elements [7]. SD modeling technique was developed by Forrester at Massachusetts Institute of Technology (MIT) in the early 60's to solve long-standing dynamic industrial management problems [8]. Today, SD is widely used to solve various business policy and strategy problems [9-11].

In SD, the "structure" of the system is defined by the totality of the relationship between the physical processes, information flows, and managerial policies. In SD, dynamic behavior patterns of a system are generated by its structure. A typical SD study focuses on understanding how the components of a system interact, how and why the dynamics of concern are generated, and then search for policies and decision rules used by upper management to improve the system performance. [11]

3 Modeling and Simulation in Cybersecurity

For analyzing complex problems such as cyber security and developing design solutions, many approaches are used in engineering science. These methods include descriptive models, system testbeds, and system (or simulation) models [12]:

- *Descriptive Cyber Security Models:* Diagrams with supporting text are used to describe a system in descriptive models. Attack graphs are example for descriptive models. A typical attack graph consists of network diagrams plus descriptions of applicable malware methods and mitigation techniques.
- *System Testbed Cyber Security Models:* System testbeds are extreme and most rigorous tools used for model analysis. These testbeds include working prototypes and live full-scale physical testbeds. Laboratory-scale equipment may be connected to sophisticated control systems to study device-level vulnerabilities. Information Warfare Analysis and Research (IWAR) Laboratory [13] is a classic example for the cyber security testbed. IWAR is an isolated laboratory for students to practice various computer security attacks/defenses.
- *Cyber Security System Models and Simulation:* System models capture the essential characteristics or behavior of the systems under study. These are middle level and lower cost methods. In this approach, generally, fully synthetic or simulated models are used for analysis and system understanding.

Though descriptive models are simple and least expensive, they do not predict the future behaviors or states of the system under study. System testbeds are very good approach for simulating technology level network attacks/defenses. But building system testbeds consume a large amount of resources, money, and time. Moreover, the system testbeds must be brought into original state before each and every cyber attack/defense run. In addition to these drawbacks, system testbeds are used to predict excessively narrow sets of problems due to the practical testbed sizes and practical limitations on approaches and measurement techniques. Therefore, the simulation model is used to better understand the behavior of the system under study or expected behavior or states of the proposed system and to study the effectiveness of the system design. [12,14,15]

When information security threats are not acute, both information security and lay managers can use modeling and simulation to better understand their information environment both on a concrete and abstract level. Once a model is developed and validated (using simulation), proactively it can be used to identify system vulnerabilities and reactively it can be used to investigate a real-world system or provide education and training by means of various "what if" questions [1,16]

Using modeling and simulation in the cyber security field provides many benefits including [4]: risk analysis, planned network change verification, security controls and resources optimization, complex network analysis, complex networks comparison, and cost-effective training to cyber security personnel.

4 Cyber Security Modeling Using System Dynamics

In SD, the system's behavior is modelled using a causal-loop diagram. The causal-loop diagram clearly indicates the linkages between the system components, the feedback loops, and the linkage between the system and its operating environment. This casual-loop diagram/analysis helps the decision-makers to understand a complex, inter-related system. SD simulation software, such as Powersim, lets the decision-makers' to extend their understanding of a system by adjusting the system parameters, linkages, feedback loops, or by rearranging components of the system. Thus, system dynamic software allows the decision makers to model wide verity of scenarios and observes the system's behaviors under these different conditions. [7]

In our proposed model, the network is considered as a system, similar to a physical system of pipes through which water flows. The amount of water that can flow into and out of node represents the bandwidth of the network traffic. A denial of service attack, for example, is modeled by trying to force more water into a node than it can handle. Another dimension of the model is the quality of the water. Network traffic that contains bogus data or viruses is thought of as water that has contaminants. The degree or type of contaminants would affect the operation of nodes and perhaps allow us to explore OSI layer 4 and above.

5 An Example System Dynamic Cyber Attack Simulation Model

Figure-1shows the part of a hypothetical University's Information Technology (IT) infrastructure network. For simplicity purpose let us assume that the IT infrastructure network consists of a learning management system (LMS) such as Canvas, a course toolkit (which is used to communicate students registered in a particular course – such as sending bulk class emails/messages), and a mail system (University email system).

In order to simulate an attack on one or more nodes in the IT infrastructure area and to study the system-level effects; that is to see what other parts of the system are affected, we have modeled the IT infrastructure system (shown in Figure 1) using system dynamics software known as Powersim. A part of the system dynamic model for Figure 1 is shown in Figure 2.

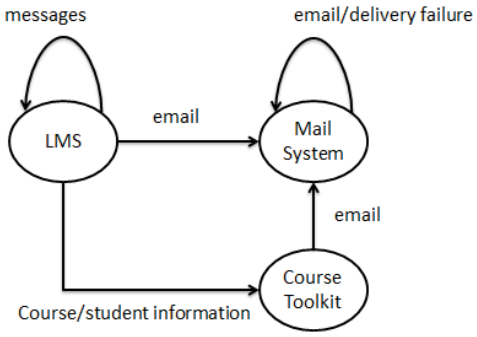


Fig.1. Part of a Hypothetical University's IT Infrastructure

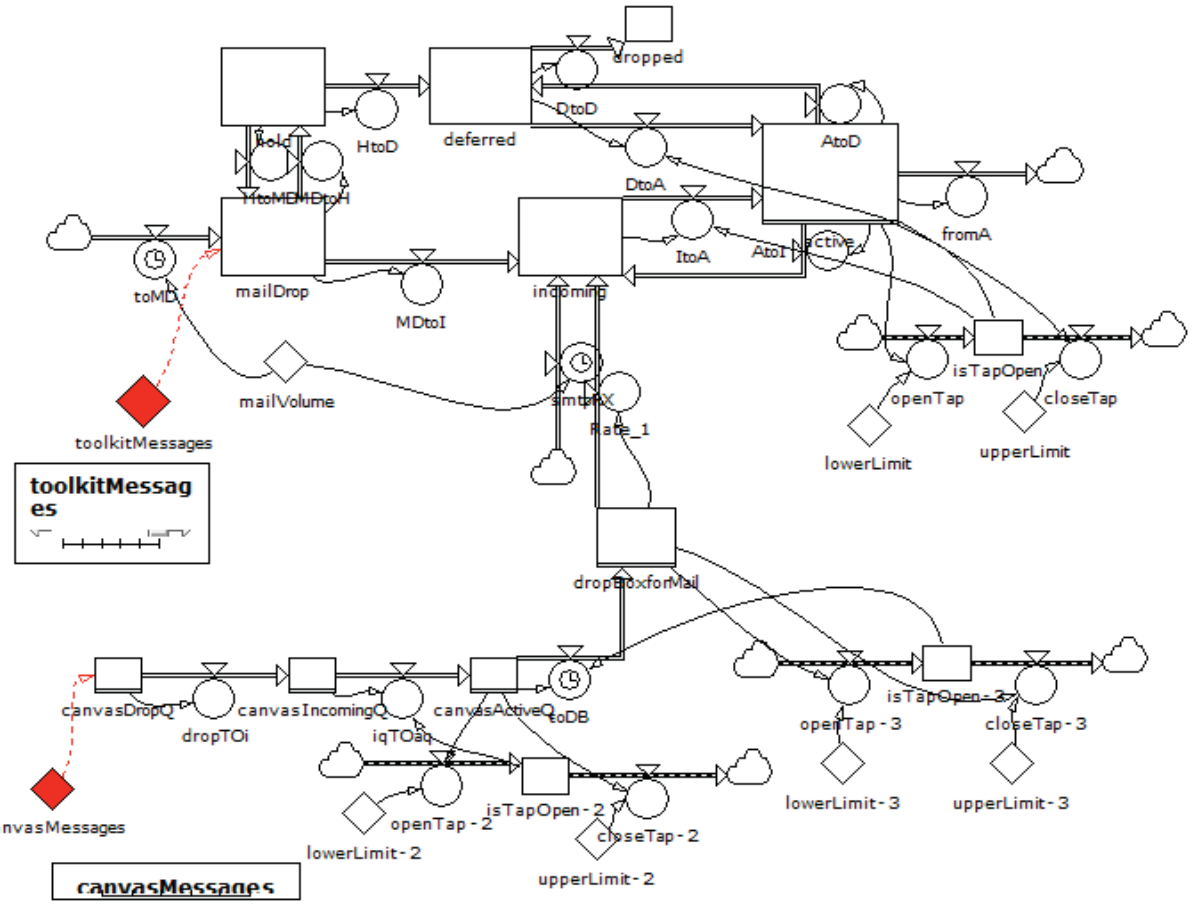


Fig. 2. The System Dynamic Model for the IT Infrastructure

The mail system(s) shown in Figure 2 consists of the following queues:

1. *Mail drop queue:* Used to hold the incoming mails from clients. The mail drop queue is normally a directory on the secondary memory in which messages can be added in offline as well.
2. *Incoming queue:* The incoming queue is analogous to the Operating Systems (OS) process ready queue. A program called pickup service will periodically scan the mail drop queue and brings the mails (if any) into the incoming queue.
3. *Active queue:* If the active queue is not full, the queue manager program will bring the new mails from incoming queue and retries of the emails from deferred queue in a round-robin fashion. Active queue is similar to the Operating Systems (OS) process run queue.
4. *Deferred queue:* The delivery failed mails will be placed on the deferred queue. Each mail in the deferred queue will be assigned a cut-off time (a time for that mail to be eligible for retry). The queue manager will scan the incoming queue and deferred queue in a round-robin fashion to transfer the mails into active queue.
5. *Hold queue:* The mails placed on hold queue stay there until either the administrator intervenes or the stayed time exceeds the maximum queue lifetime (normally 5 days).

6 Results

Let us assume that the University has 25,000 enrolled students and 25% of them are sending messages in every 30 minutes in average. Then the mail system will take approximately 10 minutes to deliver the message. This normal scenario results are shown in Figure 3.

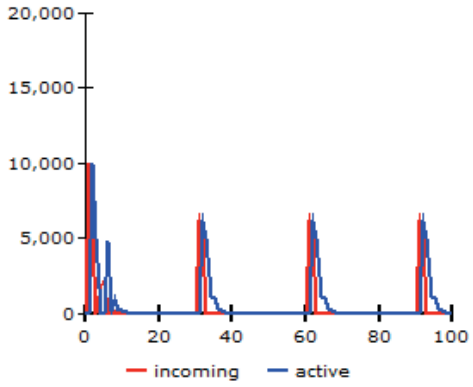


Fig.3. Mail System Queues (Normal Scenario)

The incoming queue receives approximately 6250 mails every 30 minutes and as indicated by the active queue values the mail system able to deliver all the mails within 10 minutes time. Similarly, let us assume there are 1000 messages sent by instructors and students on every 30 minutes through the LMS mail system. As shown in Figure 4, the LMS able to deliver all the messages within few minutes to the destinations.

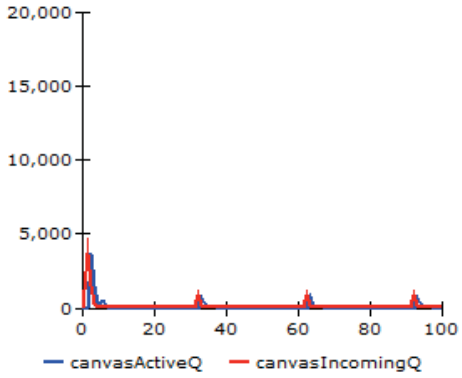


Fig.4. LMS Queues (Normal Scenario)

In majority of the cases, email system will be hosted in the University's private mail server(s) with their own unique domain name such as "@abc.edu" and the mail system in the LMS will be hosted in the LMS provider's mail servers and uses the provider's domain name such as "@instructures.com". Many mail servers, such as Postfix mail server, have a default destination concurrency limit (the default maximal number of parallel deliveries to the same destination, say 20 per hour). Whenever a mail system is unable to deliver the mail to a remote server, then it (the bounced mail) will be placed back in the deferred queue and the mail server will periodically (say every 4 minutes) retry to send the mail over a period of time (say 5 days) until it drops the mail.

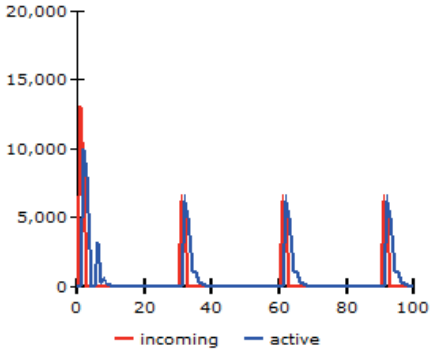


Fig.5. Mail system queues (Attack Scenario)

Let us assume an attacker inserts a bulk email (with lot of recipients) at the LMS mail system (which is hosted at "@instructures.com") with email addresses that contains the University mail server's domain name "@abc.edu". Since the LMS mail system cannot deliver more than 20 mails per hours to the same destination, in our case the bulk class emails, the remaining emails will be still waiting on the LMS

active queue and eventually causing the LMS server's active queue to be full in due course – causing Denial-of-Service (DoS); That is, when the LMS mail server's active queue is full, new legitimate users cannot send emails. The resultant DoS attack situation result is shown in Figure 6. Figure 5 and Figure 6 show the mail system and LMS queues' statuses under DoS attack.

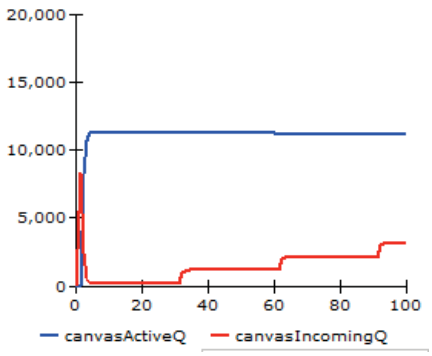


Fig.6. LMS queues (Attack Scenario)

7 Summary and Conclusion

Networks are normally modeled or simulated through discrete-event techniques. Since the primary focus of the discrete-event simulations are on packet traffic i.e., the cyber-attacks/defenses are viewed from the network layer (layer 3 in the OSI model), it obscures more insidious attacks at higher layers in the OSI model. Therefore to model cyber security attacks on host OSI layers, we have adapted a system dynamics based simulation modeling technique. In this paper we have modeled a University's information technology cyber security situation using Powersim, system dynamic modeling software, and shown the application layer Denial-of-Service attack (LMS mail system in our case). Therefore, by using known vulnerabilities, similar to this, and the current knowledge about infrastructure and security controls, the system dynamic cyber security simulation modeling allows an organization to imitate the attacker activities in OSI layer 4 and above and helps to assess and mitigate the system's risk exposure.

8 References

[1] John Saunders, "Modeling the Silicon Curtain", SANS Institute, 2001

[2] Wikipedia, "Computer simulation", https://en.wikipedia.org/wiki/Computer_simulation

[3] Romano Elpidio, Chiocca Daniela, and Guizzi Guido, "An Integrating approach, based on simulation, to define optimal number of pallet in an Assembly Line", 20th Issat Conference, Reliability and quality design, 2014

[4] "Using Risk Modeling and Attack Simulation for Proactive Cyber Security: Predictive Solutions for Effective Security Risk Management", Skybox Security Inc., whitepaper, 2012.

[5] "System Modeling and Simulation", www.inl.gov/systemsengineering

[6] Jay Wright Forrester, "Industrial dynamics", MIT Press; 1961

[7] Al Sweetser, "A Comparison of System Dynamics (SD) and Discrete Event Simulation (DES)", albert.sweetser@ac.com

[8] Barlas Y, "System dynamics: systemic feedback modeling for policy analysis in knowledge for sustainable development—an insight into the encyclopedia of life support systems", UNESCO Publishing-Eolss Publishers, 2002

[9] Coyle RG, "System dynamics modelling: a practical approach", Chapman & Hall, 1996

[10] Sterman JD, "Business dynamics: systems thinking and modeling for a complex world", McGraw-Hill, 2000

[11] Dimitrios Vlachos, Patroklos Georgiadis, and Eleftherios Iakovou, "A system dynamics model for dynamic capacity planning of remanufacturing in closed-loop supply chains", Computers & Operations Research 34 (2007) 367–394.

[12] Michael McDonald, John Mulder, Bryan Richardson, Regis Cassidy, Adrian Chavez, Nicholas Pattengale, Guylaine Pollock, Jorge Urrea, Moses Schwartz, William Atkins, and Ronald Halbgewachs, "Modeling and Simulation for Cyber-Physical System Security Research, Development and Applications", Sandia Report, SAND2010-0568

[13] Scott Lathrop, Gregory Conti, and Daniel Ragsdale, "INFORMATION WARFARE IN THE TRENCHES: Experiences from the Firing Range", Third Annual World Conference on Information Security Education (WISE3), California, USA, 2003, DOI: 10.1007/978-0-387-35694-5

[14] Dessouky, "System Simulation", lecture slides

[15] Sinclair, J. B. "Simulation of Computer Systems and Computer Networks: A Process-Oriented Approach", Rice University, 2004.

[16] Villarreal Gonzalo , De Giusti Marisa , and Texier José, "GPSS Interactive Learning Environment", Elsevier, 2012

SESSION

NETWORK SECURITY II

Chair(s)

Prof. Guillermo Francia

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

A Routing Algorithm with Path Randomization for Enhanced Security and Balanced Energy Consumption

David C. Yuan
Texas Academy of Math and Science
University of Northern Texas
Denton, Texas 76203
davidyuan@my.unt.edu

Lei Chen
Department of Information Technology
Georgia Southern University
Statesboro, Georgia 30460
lchen@georgiasouthern.edu

***Abstract*: Routers are located at the core of communication networks such as the Internet and sensor networks. The routing algorithms deployed in routers have a profound impact on network security. The traditional shortest-path algorithm used in OSPF and other routing protocols has inherent vulnerability to certain types of attacks. These routing algorithms also influence the life span of the switches and routers with high sensitivity to energy consumption. In this paper, we address these challenges by developing a novel routing algorithm with a randomization process so that packets are sent through optimal yet less predictable paths. It is expected that this process will help increase the network defense against eavesdropping and jamming attacks. It is also expected to improve the energy consumption in sensor networks and similar ad hoc networks.**

Index Terms—network security; routing algorithm; energy consumption; path randomization; wireless network;

I. INTRODUCTION

Routers and similar switching devices are located at the core of modern communication networks such as the Internet and the sensor networks. They provide the vital switching functionality so that the user packets can travel through the networks along the proper paths and eventually arrive at the aimed destinations [1]. With the rapid expansion and security penetration of communication networks, an increasing amount of high-value traffic and mission-critical data is now traveling in these networks, thus making the routers and the network traffic the prime targets of malicious attacks.

One of the most popular conventional routing protocols is the Open-Shortest-Path-First (OSPF) protocol [2]. With this protocol, the routers periodically exchange Link State Advertisement (LSA) messages with one another and learn the current network connectivity information. Using these information, a shortest path algorithm such as Dijkstra's algorithm can determine the shortest or lowest-cost path from every source node to all possible destination nodes [2][3]. The data packets then travel along these paths to reach their respective destinations.

However, this type of protocol is vulnerable to several routing threat actions, including sniffing attacks and traffic analysis [4]. Additionally, a major risk lies in the path predictability. By deploying a sniffer to eavesdrop on the LSA messages exchanged, a hacker can learn the network topology and connectivity information. Consequently the hacker can use a shortest path algorithm such as Dijkstra's algorithm to determine the shortest or lowest cost paths from every source router to all possible destination routers, just as a router does. These are exactly the same paths traveled by the user data packets as determined by the routers. As long as the hacker can gain access to one of the links on these paths, it is then possible to eavesdrop, intercept, or jam the entire communications of the intended victims. If the network topology is relatively stable, the hacker does not even need to re-calculate the paths.

To defend against such type of attacks, among many others, a number of routing protocols have deployed built-in security measures. For example, OSPF deploys a simple password protection and cryptographic authentication [5][6]. However, a weak password is often easily defeated [9]. Routers and network applications may also use Transport Layer Security (TLS) or Internet Protocol Security (IPSec) to encrypt user packets [7][8]. While the encryption technologies provide better protection, they do require more processing capabilities from the routers and user devices; and even the most powerful encryptions can still be defeated by advances in mathematics and computation capabilities [10].

Another weakness of sending all packets on the shortest or lowest cost path is unbalanced energy consumption. If a router is a part of the shortest or lowest cost path, it must process more packets than the routers that are not on such paths. This router thus must have more computational power, and/or consume more energy. For wireless networks such as sensor networks and mobile ad hoc networks where energy supply may be highly constrained, additional energy consumption by a node can shorten its service life [11].

One solution to address these challenges is to make the routing paths less predictable. Instead of using the same

shortest or lowest cost path for all traffic between a source node and a destination node, it is possible to randomly choose among multiple paths to forward the packets, thus making it more difficult for an attacker to eavesdrop or intercept the entire communication session from a victim. This scheme also spreads the routing load to more routers, thus avoids draining more power from a smaller set of routers. In order to do so, the routers should be configured such that for every destination, instead of having only one next-hop entry per destination in the routing table, there should have multiple entries for each of the alternate paths to that destination.

Some past and recent research conducted in this area include [12], [13], [14], and [15]. In [12] and [13], Luo, Liu and Fang proposed algorithms to find multiple disjoint paths at the expense of additional control messages. They assumed no topology changes during the path finding procedure. In [14], Kuo, Pang and Chan focused on Routing Information Protocol (RIP) for wired networks [16]. They proposed a path randomization algorithm that has small path-similarity sharing the minimal number of common links between source-destination nodes. The most recent work can be found in [15], where Pagan, Hession and Yuan proposed a path randomization algorithm for routing protocols such as OSPF. In their solution, instead of finding a single shortest or lowest cost path between a source node and a destination node, they ran multiple rounds of Dijkstra's algorithm. In each round, all the links belonging to the shortest or lowest cost paths found in the early rounds were excluded from being considered, thus resulted in multiple link-disjoint paths connecting the same source and destination nodes. These alternate paths were then used to create the next-hop entries in the routing table. This solution does not require any changes to LSA message content or format, and is fully compatible with existing OSPF protocol. But because the alternate paths were found running multiple round of Dijkstra's algorithm, these paths may not be optimal in term of their total length or cost. In addition, this solution may not find the disjoint paths in certain topology even if such paths exist [17].

In this paper, we propose a new routing algorithm to address the weaknesses of the solution in [15]. Instead of running multiple rounds of Dijkstra's algorithm, we propose to use both Dijkstra's algorithm and Suurballe's algorithm to find the alternate paths. Suuballe's algorithm is the optimal algorithm to find two link-disjoint paths between two end nodes [18]. It always finds the disjoint paths as long as they exist, regardless of the topology. These alternate paths are then used to populate the routing table. When a data packet arrives at a router, instead of always sending the packet by the same path as in traditional OSPF protocol, the router now randomly chooses one of the next hops for the destination in the routing table to forward the packet. Because each of the packets from a user now take a randomly chosen path at every router, it becomes much more difficult for a hacker to eavesdrop, intercept, or jam the user's entire communication session. This added layer of security is implemented at the routing layer, therefore is fully transparent to the users and network applications. It also better spreads the routing load to more routers, resulting better balance of energy consumption in a senor network or an ad hoc mobile network. The proposed algorithm is fully backward compatible with existing OSPF protocol.

The paper is organized as follows. In Section II, we describe and analyze the proposed algorithm. In Section III, we perform computer simulations to compare the algorithm with Dijkstra's algorithm and the solution in [15]. In Section IV, we conclude the paper.

II. Improved Random Paths Routing (IRPR) Algorithm

Traditional routing protocols such as OSPF use shortest or lowest cost path algorithms to determine a single optimal path connecting each source node and each destination node. One of such algorithms is Dijkstra's algorithm. The input to this algorithm is the complete network topology information, which may be obtained through LSA message exchanges with other routers. The network topology includes all the nodes, the links, and the link costs. The algorithm starts with the source node and checks all its neighboring nodes. The closest neighbor is marked. Then the source node's remaining neighbors, as well as the neighbors of the marked node(s) are checked one by one. The closest node to the source that has not been marked is now marked. This procedure continues until all connected nodes are marked. For a network with $|N|$ nodes and $|L|$ links, the running time can be as low as $O(|L|+ |N|\log|N|)$ [19].

This is a very simple and efficient algorithm to find the shortest or lowest cost paths from the starting node to every other node in a network. The security risk is that an algorithm like this is well-known. An attacker can first obtain the network topology information, and then use the algorithm to compute the exact path by which data packets will be traveling. As discussed in Section I, network traffic can be better protected if the paths taken by the packets are less predictable.

The authors of [15] proposed the Disjoint Path Routing with Random Selections (DPRRS) algorithm. In this solution, Dijkstra's algorithm is run multiple times, one for each alternate path from a source node to a destination node. The first step is to run Dijkstra's algorithm with the complete network topology. When the shortest path is generated and saved, the link cost of all the links in that path is set to infinity. This modification effectively excludes all links of the existing shortest path(s) from the alternate paths being generated later. The Dijkstra's algorithm is run again to find another path which is link disjoint from the first one.

This process repeats until the desired number of disjoint paths is generated, or until a new disjoint path can no longer be found. For a network with $|N|$ nodes and $|L|$ links, the running time is $O(kL+ kN\log N)$ for k alternate paths.

However we see two major weaknesses in this algorithm. First, the total length or cost of the k alternate paths generated may not be the minimum. For instance, when k = 2, it has been proven that Suurballe's algorithm always generate the minimum total length or cost whereas the two-step DPRRS algorithm in this case does not. The second weakness is, for certain network topology, the DPRRS algorithm may not find the disjoint paths, even if they exist. For instance, when k = 2, the DPRRS algorithm cannot find the two disjoint paths *s*-*e*-*b*-*d* and *s*-*a*-*f*-*d* between node *s* and node *d* as shown in Fig. 1, while. Suurballe's algorithm can.

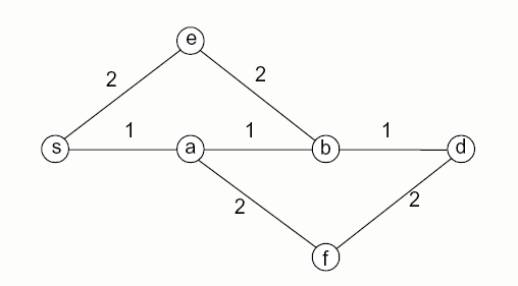


Fig.1. A network with two disjoint paths [17].
The numbers indicate link costs.

The details of our proposed algorithm are given below:

Name:	Improved Random Paths Routing (IRPR) Algorithm
Inputs:	Network $G(N, L)$ where N is the set of nodes and L is the set of links with a cost c_l associated with each link, assuming all links are bidirectional; Node *s*: source node; Node *d*: destination node; *k*: the number of link-disjoint paths connecting *s* and *d* to be found and $k > 1$.
Output:	Up to *k* link-disjoint paths connecting *s* and *d*

Pseudo code:	$p = k$;
	while $p \neq 0$, repeat: //loop begins
	if $p = 1$,
	execute Dijkstra's Algorithm with $G(N, L)$ and *s*, *d* as input;
	if succeeds,
	save the returned path;
	$p := 0$;
	else
	break the loop;
	else
	execute Suurballe's Algorithm with $G(N, L)$ and *s*, *d* as input;
	if succeeds,
	save the two returned paths, and
	set the cost of all the links in the two disjoint paths to infinity;
	$p := p - 2$;
	else
	break the loop;
	//end of loop
	return *k*–*p* and all the saved paths;

The major component of this algorithm is Suurballe's algorithm. If the algorithm runs successfully, it executes $\lfloor k/2 \rfloor$ rounds of Suurballe's algorithm and only $k\%2$ round of Dijkstra's algorithm ($\lfloor\ \rfloor$ represents floor operation and % represents modulus operation), and returns k link disjoint paths. Because the running time for Suurballe's algorithm and Dijkstra's algorithm are $O(|N|^2\log|N|)$ and $O(|L|+|N|\log|N|)$ respectively, the total running time for the IRPR algorithm is at most $O(|L| + k|N|^2\log|N|)$, which is polynomial to k, $|L|$ and $|N|$.

Compared to the DPRRS algorithm proposed in [15], our IRPR algorithm has longer running time, due to the complexity of Suurballe's algorithm. However, the IRPR algorithm has a higher probability of finding disjoint paths, and the disjoint paths it finds potentially have shorter total length or lower total cost. These are important improvements because being able to find more disjoint paths leads to more paths for a router to randomly choose from when forwarding user packets, thus improves security and achieves more balanced energy consumption among the routers. Shorter total length or lower total cost of the disjoint paths also have the advantage of requiring less network resource allocation, and users packets may experience shorter delays.

III. Computer Simulations and Analysis

In this section we conduct simulations to compare the performance of our IRPR algorithm with the DPRRS algorithm and the traditional Dijkstra's algorithm. Using LEDA programs, a C++ class library for efficient data types and algorithms, we generated network graphs with network sizes of 10, 20 and 40 nodes and an average nodal degree (i.e., the number of links ending in a node) of 2.8 for testing [15] [20]. We run the IRPR algorithm to generate 2, 3, and 4 link-disjoint paths for every pair of source-destination nodes,

and compare the results with those of DPRRS listed in [15]. Based on the desired number of disjoint paths, we label each variation of the IRPR algorithm as IRPR-2, IRPR-3 and IRPR-4, similar to how the DPRRS algorithm are labeled in [15] as DPRRS-2, DPPRRS-3, and DPRRS-4.

We are most interested in two matrices, the success rate of finding the desired number of disjoint paths, and the total length or cost of the disjoint paths being generated. For the path length or cost, we agree with [15] on that the most significant network resource consumptions and delays occur at the routers; therefore we set all link cost to 1 when we ran the algorithms, which made the path length equal to the hop count. A hop count is the number of routers on a path connecting the source node and the destination node.

For all source-destination node pairs, we ran the three variations of the IRPR algorithm, and compare its success rates and the average hop-counts with those of the DPRRS algorithm and the Dijkstra's algorithm.

We observed that the success rates for the IRPR algorithm and the DPRRS algorithms were nearly the same in all the simulations we performed. This seemingly surprising result may indicate for networks of nodal degree of 2.8 or higher, there exist sufficient numbers of alternate paths between a source node and a destination node; hence both Suurballe's algorithm and two-step Dijkstra's algorithm are equally likely to find these paths.

For the average hop counts of the disjoint paths, we first compare the results of IRPR-2, IRPR-3, IRPR-4 and Dijkstra's algorithm. The comparisons are depicted in Fig 2. As expected, the single shortest paths generated by Dijkstra's algorithm have the lowest average hop counts. For the IRPR algorithm, the more disjoint paths it must generate, the higher the average hop counts become. This is because the additional paths must take longer routes in order to be disjoint from each other. It can also be observed that the more nodes a network has, the longer the paths becomes, simply because larger networks provide more connectivity; therefore there are increased path variations and the average distances between the nodes become larger.

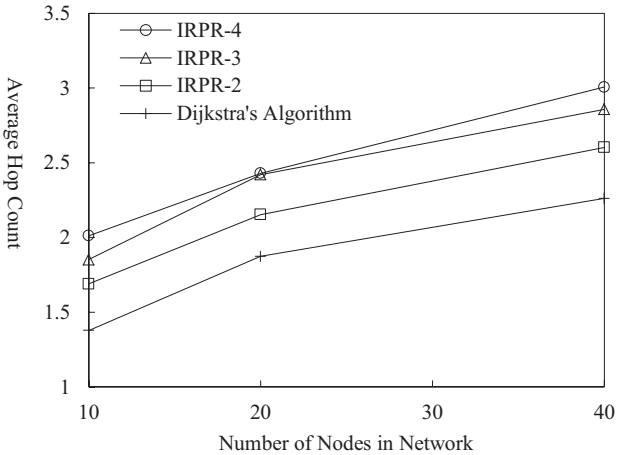


Fig 2. Comparison of the average hop counts using variations of the IRPR algorithm and Dijkstra's Algorithm

Next we compare the average hop counts for the same number of disjoint paths generated by the IRPR algorithm and the DPRRS algorithm. The results are depicted in Fig 3, Fig 4, and Fig 5. It is very clear that the IRPR algorithm consistently and significantly outperforms the DPRRS algorithm for all network sizes and all numbers of disjoint paths generated. To be more specific, for 2 disjoint paths, IRPR outperforms DPRRS by 11.62% to 15.56%; for 3 disjoint paths, IRPR outperforms DPRRS by 14.97% to 19.04%; and for 4 disjoint paths, IRPR outperforms DPRRS by 19.20% to 22.99%. These results match our expectations discussed in Section II. It also appears that the more disjoint paths are to be generated, the more advantages the IRPR algorithm become over DPRRS.

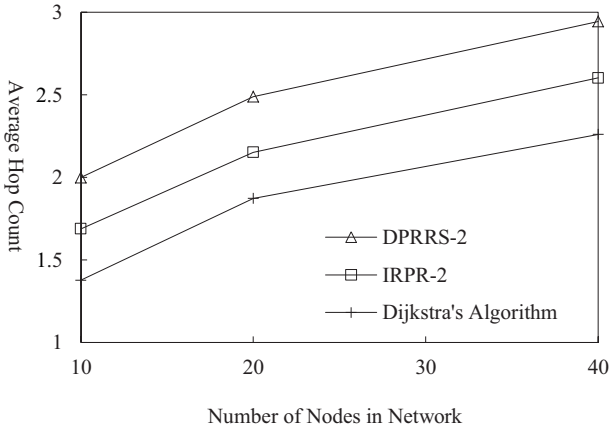


Fig 3. Comparison of the average hop counts of two disjoint paths using the IRPR algorithm and the DPRRS algorithm

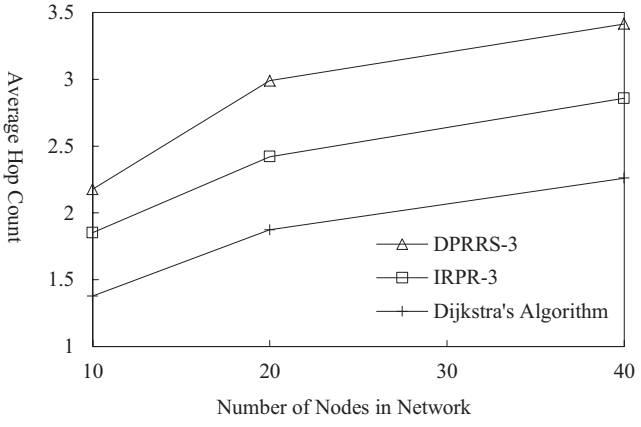


Fig 4. Comparison of the average hop counts of 3 disjoint paths using the IRPR algorithm and the DPRRS algorithm

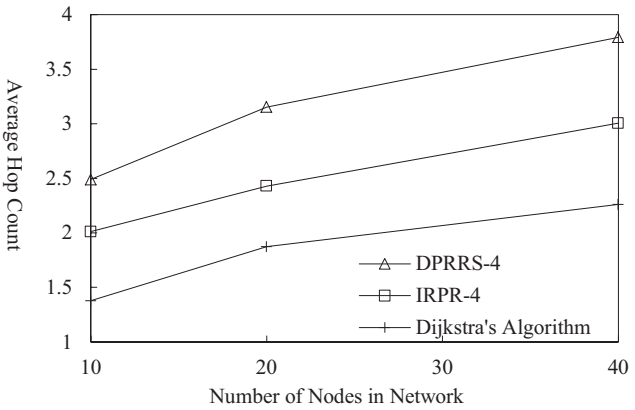


Fig 5. Comparison of the average hop counts of 4 disjoint paths using the IRPR algorithm and the DPRRS algorithm

IV. CONCLUSION

Traditional routing protocols such as OSPF use shortest path or lowest cost routing algorithms to determine the optimal path for a given source node and a destination node, which may lead to various security vulnerabilities, and unbalanced energy consumptions by the routers. In this paper, we proposed a new routing algorithm, IRPR that generates multiple disjoint paths for the routers to randomly select from. The algorithm's complexity is polynomial and can be easily implemented. Computer simulations confirmed that the algorithm consistently and significantly outperformed the existing solutions in term of path hop counts. For future study, we will investigate more advanced solutions, especially when more disjoint paths are desired and when node disjoint paths are desired.

REFERENCES

[1] Douglas Comer, *Internetworking with TCP/IP*, 6th Edition. Pearson.
[2] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*, 1st Edition. Addison-Wesley Professional.
[3] N. Jasika, N. Alispahic, A. Elma, K. Ilvana, "Dijkstra's Shortest Path Algorithm Serial and Parallel Execution Performance Analysis." *Proceedings, IEEE 35th International Convention MIPRO*, 2012. Pages: 1811-1815.
[4] A. Barbir, S. Murphy, and Y. Yang. "Generic Threats to Routing Protocols." *https://www.ietf.org/rfc/rfc4593.txt*
[5] Vetter, Brain, Feiyi Wang, and S. Felix Wu. "An Experimental Study of Insider Attacks for OSPF Routing Protocol," *Proceedings, IEEE International Conference on Network Protocols*, 1997. Pages: 293-300.
[6] Fang Ying-lan, Han Bing, and Li Ye-bai. "Research and Implementation of Key Technology Based on Internet Encryption and Authentication." *Proceedings, IEEE International Conference on Networking and Digital Society*, 2009. Pages: 179-182.
[7] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol," *https://tools.ietf.org/html/rfc5246.*
[8] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," *https://tools.ietf.org/html/rfc4301.*
[9] Kamal, S., and B. Isaac. "Analysis of Network Communication Attacks," Proceedings, *IEEE 5th Student Conference on Research and Development*, 2007. Pages: 1-6.
[10] Charlie Kaufman and Radia Perlman. *Network Security: Private Communication in a Public World*, 2nd Edition. Prentice Hall.
[11] Yi-Bing, Li Hai-Bo, Li Zhong-Cheng, and Dutkiewicz, E. "A New Method of Selecting Stable Paths in Mobile Ad Hoc Networks." *Proceedings, IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006. Volume: 2. Pages: 38 – 45.
[12] Wenjing Lou, and Yuguang Fang. "A Multipath Routing Approach for Secure Data Delivery." *Proceedings, IEEE Military Communications Conference, MILCOM* 2001. Volume: 2. Pages: 1467 – 1473.
[13] Wenjing Lou, Wei Liu, and Yuguang Fang. "Spread: Improving Network Security by Multipath Routing." *Proceedings, IEEE Military Communications Conference, MILCOM* 2003. Volume: 2. Pages: 808 - 813.
[14] Chin-Fu Kuo, Ai-Chun Pang, and Sheng-Kun Chan. "Dynamic Routing with Security Considerations". *IEEE Transactions on Parallel and Distributed Systems*. January 2009. Pages: 48-58.
[15] Mario Pagan, Audrey Hession, and Shengli Yuan. "A security-enhanced routing algorithm with path randomization." *Proceedings, IEEE International Conference on Computing, Networking and Communications* (*ICNC*) 2015. Pages: 1137 – 1141.
[16] G. Melkin. "RIP Version 2." *https://tools.ietf.org/html/rfc2453*
[17] Shengli Yuan and Jason P. Jue, "Dynamic Lightpath Protection in WDM Mesh Networks under Wavelength-Continuity and Risk-Disjoint Constraints," *Computer Networks Journal* (*Elsevier*), vol. 48, no. 2, pp. 91-112, June 2005.
[18] J. W. Suurballe, R.E. Tarjan, "A quick method for finding shortest pairs of disjoint paths." *Networks*. Vol. 14. 1984. Pages: 325-336.
[19] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithms*, McGraw Hill, 1997
[20] Algorithmic Solutions Software GmbH, *http://www.algorithmic-solutions.com/leda/index. htm*

Federated Identity and Access Management and Trusted Computing-based Federated GRID Model for Federated GRID Resources

Zubair Ahmad Khattak[1], **Jamalul-lail Ab Manan**[2], **Suziah Sulaiman**[3]
[1]Department of Computer Science, IQRA National University, Peshawar, KPK, Pakistan
[2]MIMOS Berhad, Technology Park Malaysia, 57000, Kuala Lumpur, Malaysia
[3]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Perak, Malaysia

Abstract - *The prominent FId model allows the IDP end-users to log-in once via a service SSo to access multiple resources using SAML or XACML at the RP's. The GridShib, Shibboleth plus PERMIS, Globus Toolkit, and PERMIS are the examples of web-portal based GRID. The problems in the existing web portal-based GRID are that: (1) The protected resource access decisions are performed on attributes like name, email, or role, but not on the attributes of the attested machines', and (2) The RP blindly trusts the IDP machine's health. In this paper the conceptual federated GRID model is proposed by taking advantage of the TC, and FId&AM systems. The contributions in this paper are: (1) the lma and rma protocols for the federated GRIDs, (2) the resources access decision on the basis of attested machine's platform mutual integrity attribute, and (3) the machine-platform trust formation via the machine's platform integrity mutual attestation.*

Keywords: attributes-mutual attestation; federated identity and access management; mutual remote attestation; GRID; trusted computing

1 Introduction

The FId&AM systems facilitate the participating organisations in AuthN, AuthR, sharing identity attributes, and granting/denying access to the protected resources using federated standards (e.g., SAML and XACML). The FId&AM systems consist of three main entities: (1) IDP, (2) SP, and (3) end-users. The IDP organisation is liable to generate and retain the end-users credentials and attributes, and the SP organisation is liable to release a protected resource to the legitimate IDPs end-users.

The Hard-Security processes, such as AuthN and AuthR, are used in FId&AM systems to protect the resources' unauthorised abuse [1]. The traditional AuthN credentials solve two concerns related to an ID: (1) the end-user's legitimacy, and (2) the end-users "ID" or "account ownership". The FId&AM systems AuthR process follows the successful execution of the AuthN process. In the AuthR process, the resource access decisions (i.e., allow or deny) are performed on the "attributes sets". The two types of attributes are static and dynamic. The benefit of having "dynamic/variable attributes" over "static attributes" are: (1) make use of the "name" attribute at the SP_1, (2) make use of the "name+email" attributes at the SP_2, & (3) make use of the "role" attributes at the SP_3.

In GRIDs the renowned PKI scheme is used to authenticate and authorise the end-user to access services. To validate the GRID end-user's authenticity, he/she must have a legal PKI certificate and a corresponding password. In the GRIDs "ID trust" deals with the end-user's certificate genuineness validation and resources' access decisions (allow or deny). The techniques used to establish "ID trust" are AuthN protocols and access control methods. The main issues with the PKI-based certificate attributes are inflexibility/stationary (e.g., a role change leads to attribute alteration), and privacy (e.g., super certificate reveals more information to SPs). As a result, for the federated model, a scalable solution is useful. In this model: there is an IDP to validate the end-user's authenticity; AA provides the attributes to the SP; and the SP allows/denies a resource to the end-user. Another class of trust is the "behaviour trust" dealing with the "trustworthiness" of a client's or server's machine integrity. The later "class of trust" is the core of this work. The GRID's traditional AuthN mechanisms are not appropriate to report "the end-user's client machine health status (i.e., its integrity)" to the server (i.e., resource provider) that "I (client) am in a trustworthy or honest state" and vice versa.

In federated GRID setup, each GIDP is responsible for its own end-users AuthN, AuthR, identity attribute sharing, and GSPs to process grant/deny decisions to the protected resources. The six (6) major federated GRID entities are: (1) *End-user* (the grid resource consumer), (2) *GRID machines* (the end-users and GIDP), (3) *GSP* (performs resource access decisions), (4) *GIDP* (responsible for the end-user AuthN), (5) *GVOs* (the pool of GRID's participating virtual organisations (allows faster data access for later retrieval)), and (6) *GAA* (handles the attributes request/response queries).

ISBN: 1-60132-445-6, CSREA Press ©

This paper discusses the major question of "*How can one access federated GRID web-portal resources, securely, in a trustworthy mode*?"

TC [3] hardware-rooted security solution, TPM [4] and the *rma* scheme can be used to build a secured and trustworthy federated GRID resource access ecosystem.

The main contributions in this paper are as follows:

- The *lma* and *rma* protocols for the federated GRIDs.
- The *rma* protocol integration with the basic ID and PWD AuthN scheme for federated GRID web-portal resources access.
- The federated GRID's resource access via the machine's mutual attestation attributes.
- The comprehensive AuthN, AuthR, and machine's mutual integrity protocol for the federated GRID.

This paper discusses the background in Section 2, existing web-portal GRID's solutions in Section 3, and security threats in Section 4. Next, Section 5 discusses the proposed solution. The second to last, Section 6, discusses the discussion, and finally, Section 7 discusses the conclusion and future work.

2 Background

2.1 Trusted Computing

The TC technology is an initiative of the Trusted Computing Platform Alliance (TCPA), later the name was changed to TCG [3]. The twofold advantages of TC are: (1) TC technology will make the computing nodes safer, add trustworthiness to the computing platform and be less prone to malwares and viruses, and (2) the TC technology will enhance the computing nodes' security by a chip TPM. TPM is a micro hardware chip, secure cryptographic processor, which integrates cryptographic functionalities inside the TPM chip [4]. TPM hardware chip provides different capabilities, such as the secure generation of cryptographic keys, RNG (Random Number Generator), Binding, Sealing, and RA [5]. The EK, a unique "RSA key", private and public key parts of the EK do not leave the TPM boundaries for privacy reasons. Therefore, in place of the EK, the AIK is used to not reveal the TPM's EK outside of the TPM.

2.2 Remote Mutual Attestation

In the *rma* scheme, "the challenger device challenges the remote target device's platform integrity status to validate its honesty or dishonesty and vice versa". The drawback of the TCG RA scheme is that "the device's target and challenger are powerless to validate, mutually, each other's platform integrity status". Another common drawback of the TCG RA approach is that it only measures the BIOS, boot loader, etc. loaded prior to an operating system. To overcome the issues in the TCG RA scheme, Sailer et al. [17] came up with an approach known as the IMA that consists of three major components: (1) integrity measurement mechanism, (2) integrity defy (challenge) mechanism, and (3) integrity corroboration (validation) mechanism. The first component is liable for deciding what to measure, and to securely keep the measurements. The second component is liable for machine platform's measurement lists retrieval and freshness verification. The third component is liable to validate the completeness of the measurements list, un-tampered, and unsullied. In this work TPM-based mutual integrity protocol-based using IMA approach is proposed [17].

2.3 Security Assertion Markup Language

The concept of the federated identity between business partners is turned into a reality because of the SAML standard [6]. The SAML was developed by OASIS and SSTC. The Internet2 Shibboleth project, Liberty Alliance, and OASIS WS-SecTC (Web-Services Security Technical Committee) have all adopted the SAML. The SAML works with all major protocols, such as SMTP, HTTP, SOAP, and FTP.

2.4 Extensible Access Control Markup Language

The XACML [7], an OASIS standard, is an ABAC language to use several attributes in the AuthR process. The PDP, PAP, PIP, and PEP, PIP are the main XACML points that take part in the subject action on a resource. The basis of the XACML policies are: a subject, an action, and a resource. For instance, a clinic head nurse (subject) may update (action) some files (resource).

3 Existing Web-portal Grid's Progress

The GTkit [8], PERMIS [9], Shibboleth [19] with PERMIS [10], and GridShib [11] are the well known existing GRID solutions. In the following, these different web-portal based GRID schemes are discussed:

3.1 GridShib (Shibboleth)

Project-GridShib, initiated in 2004, resolves the AuthN, AuthR, and interoperability concerns between Shibboleth and the GTkit [16]. The focal-point of project-GridShib was to power-up the Shibboleth AMI (Attribute Management Infrastructure) by carrying attributes of Shibboleth as SAML-assertions from the Shib-IDP to service the PDP, a decision end of the GTkit. In this approach, the end-user posses a legitimate PKI certificate [2]. The seized certificate needs to have the end-user's institution name, i.e., the IDP. In the next step, the seized certificate is handed-over to the module-application, and after successful verification, the information of the IDP is hauled-out. Finally, the application transfers the certificate ID to the IDP, and the application, in return, receives the end-user attributes in a SAML assertion. Welch et al. [14] provided integration of the GTkit+Shibboleth to enable attribute-based AuthR and pseudonymity.

3.2 Shibboleth plus PERMIS

PERMIS expands Shibboleth's AuthR with a precondition verdict and role hierarchies. In the Shibboleth

architecture, the SP trusts the entity's a.k.a IDP. PERMIS can either use Shibboleth or Apache to provide end-user AuthN. In PERMIS+Shibboleth, the integration of the apache module-PERMIS SAAM (Shibboleth Apache AuthR Module) makes use of PERMIS to manage access to digital websites. PERMIS can make complicated decisions because PERMIS' PDP and PV are both policy centric, which means it can check random conditions prior to granting access. PERMIS is an example of an RBAC, AuthR system, which uses XML written policies [12], [13]. Figure 1 depicts steps 1 to 12 of "how the end-user can request a remote protected resource via the features of Shibboleth plus PERMIS".

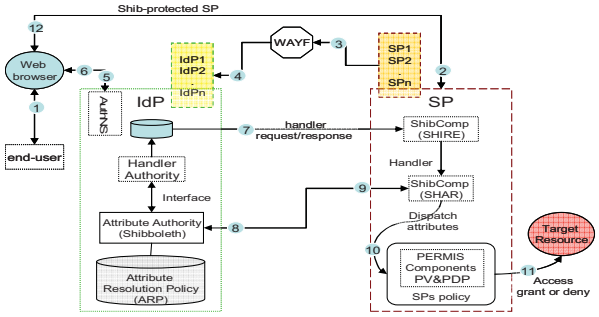


Figure 1. Shibboleth plus PERMIS target resource access resolution

3.3 Globus Toolkit (GTkit)

The GTkit [8] is an open source GRID software implementation developed by the GLOBUS Alliance. The GLOBUS coalition is a group of various organisations and people developing a range of technologies appropriate for GRID computing. The major GTkit components are runtime components, security components, data management components, etc. Also, the GTkit is a combination of several tools which make it easier to deploy an emerging GRID resource (or service). GTkit4, GTkit3, and GTkit2 are different versions of the GTkit. GTkit3.3 and GTkit4 support SAML and XAML, respectively. GTkit3.3 and onward versions pool SAML and XACML, respectively, with PERMIS for the purpose of the AuthR process.

3.4 PERMIS

PERMIS is an an AuthN system. PERMIS makes use of the RBAC model standard of the NIST (National Institute of Standards and Technology) [15]. PERMIS is an AuthR (or PrivilEge management) system that controls access to an organisation's resources (or services) based on the roles of the end-user and access control polices. The strength of PERMIS is its integration ability into AuthN schemes, such as Shibboleth, UN+PWD, PKI, Grid Proxy certificate, and Kerberos.

4 Threats

Bhatia [20] discussed different GRID security risk questions and critical security problems, such as the AuthN, AuthR, and access mechanisms. Khattak et al. [21] discussed ID theft and the weak trust association threat model for the federated schemes of which the threat model is also applicable to the GRID ecosystem.

Table I shows that the major threats to the distributed (or federated) GRID's end-users and IDP machines are computer Trojans, and rootkits, etc. The infected GRID's clients' (or IDPs) machines with Trojans and rootkits can behave maliciously/dishonestly and allow the intruders or malicious program code to harm the GRID's machines.

TABLE I. THREATS AND SECURITY CONCERNS TO GRID'S MACHINES

Threats	Security Concerns
Trojans	A malevolent program may infect EUm or GIDPo machines to perform harmful or unwanted functions.
Rootkits	The intruder can gain an administrative level control of the rootkit infected machines' EUm and GIDPo.
Replay attack	A malevolent attesting machine may replay EUm and GIDPo machines' attestation info (log+TPM sum).
Masquerading attack	A malevolent attesting machine (or intermediate invader) may substitute the EUm and GIDPo machines' genuine log+TPM sum.
Tampering attack	A malevolent attesting machine (or intermediate invader) may tamper with the EUm and GIDPo machines log+TPM sum.

The mutual integrity protocol must protect the EU_m and $GIDP_o$ machines attestation information against different threats, such as replay attacks, masquerading, and tampering.

To assure the requirements of confidentiality and authenticity of the end-user's AuthN credentials and machine's mutual integrity attestation mechanism, the authors assume that the HTTPS (SSL/TLS) is in-place.

5 Proposed Solution

The distinction between the lma and *rma* are as follows:

5.1 Local and Remote Mutual Attestation

In the lma scheme (Figure 2), the EU_m machine's and the $GIDP_O$ machine's mutual attestation (steps 5&6) is carried-out at the local machines (i.e., attest their platform recorded integrity with the local validation repository). In the lma scheme, for the successful execution of steps 5&6, the EU_m AuthN is mandatory.

In the *rma* scheme (Figure 3), the EU_m machine's and the $GIDP_O$ machine's mutual attestation (steps 5&6) is carried-out remotely on the $GIDP_O$ machine. Similarly to the lma in the *rma* scheme, the EU_m's successful AuthN is mandatory. The *rma* scheme can utilise any kind of attestation method (i.e., IMA [17], PBA [18], etc.). For the purpose of this paper, the mutual integrity challenge protocol based on IMA [17] is used (combined with the UN/PWD) in a federated GRID ecosystem that is "how the GRID's EU_m and $GIDP_o$ machines' mutual integrity is validated/ verified".

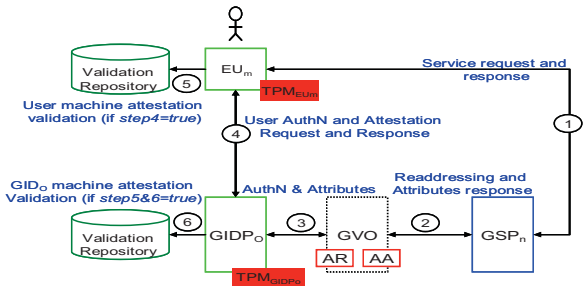


Figure 2. Local mutual attestation scheme

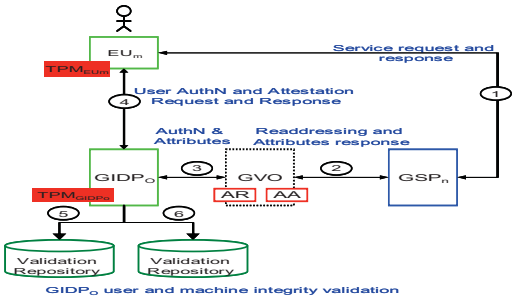


Figure 3. Remote mutual attestation scheme

5.2 Proposed Federated GRID Model

Figure 4 shows the: (1) EU_m AuthN and (2) GRID machines' ($GIDPo_{(IDp)}$ and EU_m) attestation. The $GRDPo_{(IDp)}$ verifies the EU_m credentials and AA is responsible to decide the kind of attributes to be released to the GVO using AR policies achieved by SAML (method-post OR artifact). In the case of ID_m/PWD_m, the *SAMLAuthNMethod* is put to "*BASIC*". The ACUs at the EU_m and $GIDPo_{(IDp)}$ process the incoming and outgoing attestation SAML request/response. The nonce's N1/N2 enables protection against replay attacks.

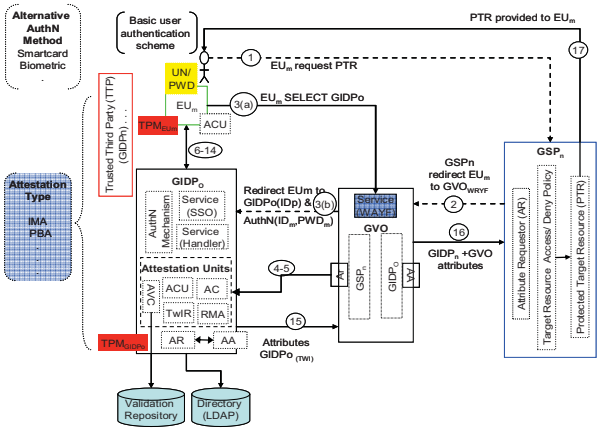


Figure 4. FId&AM and RMA-based federated GRID web-portal resources access model

The comprehensive AuthN, AuthR, and machines mutual attestation protocol interaction (Figure 5 (page 5)) for federated GRID model is discussed in the following:

5.2.1 The end-user (EU_m) authentication

- Protected resource request (step-1).
- Entity GSP_n redirect EU_m to GVO $Service_{WRYF,}$ produces a virtual cookie (vc) and records the address of GSP_n (step-2).
- The EU_m selects the $GIDP_{O(IDp)}$ entity from the IDPs pool. The EUm is then redirected to the $GIDP_{O(IDp)}$ to login with the provided IDm/PWDm. The provided IDm and PWDm are then validated with the LDAP stored entries at $GIDPo_{IDp}$ (step-3 (3(a) and (3(b)).

5.2.2 The attribute collection

- The AA then contacts the Ar to collect the attributes (step-4).
- The TwIR then contacts the rma-component for some attributes (step-5).

5.2.3 The EU_m & $GIDP_o$ remote mutual authentication

- To attest the EUm & $GIDPo_{(IDp)}$ machines (step-6).
- The AC makes an attestation query+160-bit random N1 to the ACU of the EU_m (step-7).
- The ACU then collects the EUm machine token and forwards it via AC to the AVC located at the GIDPo. The AVC then fetches the AIKpub, validation of the TPMsig, nonce N1 is matched with the nonce N1 in step-7, re-calculates the sml tpm aggregate and compares it with the tpmQuote PCR[v] (step-8).
- The AVC then forwards the attestation outcome (AO=True) of EU_m to the $RMA_{GIDPo(IDp)}$ (step-9).

5.2.4 The $GIDPo_{(IDp)}$ machine attestation

- To attest the $GIDPo_{(IDp)}$ machines (step-10).
- The AC makes an attestation query+ 160-bit random nonce (N2) to the ACU (step-11).
- The ACU then collects the $GIDPo_{(IDp)}$ machine's token and forwards it via AC to the AVC located at the $GIDPo_{(IDp)}$. The AVC then fetches the AIKpub, validation of TPMsig, nonce is matched with the nonce in step-11, re-calculates the sml tpm aggregate, and compares it with the tpmQuote (step-12).
- The AVC then forwards the attestation outcome (AO=True) of $GIDPo_{(IDp)}$ to the $RMA_{GIDPo(IDp)}$ (step-13).
- The $RMA_{GIDPo(IDp)}$ passes the attestation outcome ="True" to the TwIR (resolver-TwowayIntegrity) located at $GIDPo_{(IDp)}$ (step-14).
- The TwIR returns the "attribute-twi" to the GVO (step-15).

5.2.5 The EUm redirection, back to the GSP_n

- The EUm is redirected by the GVO to the GSPn whose ID ($GSPn_{IDq}$) was recorded earlier with the GVO (step-16).
- The protected resource (R_p) is provided to the EU_m (step-17).

1) $EU_m \rightarrow GSP_n$: R_P
2) $GSP_n \rightarrow GVO_{[Service\text{-}WRYF]}$: vc, $GSPn_{IDq}$
3) $EU_m \rightarrow GIDPo$: $GIDPo_{IDp}$
   : $ID_m \backslash\backslash PWD_m$
4) $AA_{GIDPo(IDp)} \rightarrow Ar_{GIDPo(IDp)} : A_r$
5) $TwIR_{GIDPo(IDp)} \rightarrow RMA_{GIDPo(IDp)}$: A_{twi}
6) $RMA_{GIDPo(IDp)} \rightarrow AC_{GIDPo(IDp)}$: EU_m
7) $AC_{GIDPo(IDp)} \rightarrow ACU_{EUm}$: $Att_{Req,}$ N1
8) $ACU_{EUm\ (TPM)} \rightarrow AVC_{GIDPo(IDp)}$
   : $Quote_{sig\ ([PCR\ [v],\ N1)AIKprv],}$ sml
   : [*cert*(AIKpub), *sig*[(PCR[v] + N1)*AIKprv*], (N1, sml using PCR)]]
9) $AVC_{GIDPo(IDp)} \rightarrow RMA_{GIDPo(IDp)} : AResult_{EUm\ (true)}$
10) $RMA_{GIDPo(IDp)} \rightarrow AC_{GIDPo(IDp)}$: $GIDPo_{(IDp)}$
11) $AC_{GIDPo(IDp)} \rightarrow ACU_{GIDo(IDp)}$: $Att_{Req,}$ N2
12) $ACU_{GIDPo(IDp)\ (TPM)} \rightarrow AVC_{GIDPo(IDp)}$
   : $Quote_{sig\ ([PCR\ [v],\ N2)AIKprv],}$ sml
   : [*cert*(AIKpub), *sig*[(PCR[v] + N2)*AIKprv*], (N2, sml using PCR)]]
13) $AVC_{GIDPo(IDp)} \rightarrow RMA_{GIDPo(IDp)} : AResult_{GIDPo(IDp)\ (true)}$
14) $RMA_{GIDPo(IDp)} \rightarrow TwIR_{GIDPo(IDp)}$: $AResult_{GIDPo(IDp)\ (true)}$
15) $TwIR_{GIDPo(IDp)} \rightarrow GVO$: twi
16) $GVO \rightarrow GSPn_{(IDq)}$: twi
17) $GSPn_{(IDp)} \rightarrow EU_m : R_p$

Figure 5. Federated GRID model comprehensive AuthN, AuthR, and machine mutual attestation protocol

6 Discussion

In reliasing a web-portal based federated GRID model for resources, the following lessons have been are learned:

6.1 SAML and XACML authorisation process

Suppose that the "EUm identity (ID_m)" and the honesty of the "$GIDP_{O(IDp)}$ and EU_m machines" are validated successfully and the requested attributes are accessible to the GSP_n.

In the case of the *RBAC,* the entity-*GVO* AA passes on the "SAML-assertion" to the web-app "xyz".

For instance, David, a junior nurse, can only "read" the patient's personal information on "xyz" but cannot read the patient's "infection history", and John, a head nurse, and Sara, a doctor, can "read and edit" all of the patient's information on the "xyz".

In the case of the *XACML,* the EU_m contacts the PDP, and if the EU_m has positive access rights, then it updates all the services.

For instance, the EU_m of a domain "X" wants to use, remotely, a resource, "Statistical Analysis Tool (SAT)", he/she 1st converses with the point k.a. "PDP" at the GSP_n "Controller". In this case, the necessary regulations let the EU_m to execute the SAT, remotely.

6.2 Tackling threats in federated GRIDs

6.2.1 Rootkits

The rootkits have a quality that allows them to modify the usual programs and system's libraries on the EUm and GIDPo machines. The imposter would gain an unauthorised administrative level access to the EUm and GIDPo machines to abuse the GRID resources.

6.2.2 Replay attack

In both the EUm and GIDPo machines' attestation cases, the organisation GIDPo machine has the authority to verify the originality of the TPMQuote and PCR (tpm_aggregate). In both cases, their originality is assured by the two nonces (N1 &N2) being identical (steps 8 & 12). The nonces inclusion in the EUm and GIDPo attestation assures that the EUm and GIDPo attestation reply is the latest and not the former one.

6.2.3 Masqurading attack

The component at the GIDPo recovers (steps 8 & 12) the certificates (cert(AIKpub)) of EUm and GIDPo machines which bind the AIKpub (a proof key) of the TPMQuote to a specific federated GRID domain's particular systems. In our model, comparing the EUm and GIDPo machines' irreplaceable identification with the corresponding identification in their certificates (cert(AIKpub)) uncovers the masquerading attack.

6.2.4 Tampering attack

In steps (8&12), the signature validation, sig[(PCR[v]+N1)AIKprv], for the EUm and GIDPo machines, can assist to detect the tampering attack with the tpm_aggregate. The tampering attack on the EUm and GIDPo machines' smls is detectable by inspecting the received smls, recalculating the tpm_aggregate via a tpm_extended operation, and the results are compared with the tpm_aggregate PCR[v] (part of a signed TPMQuote). In the federated GRID model, if the EUm and GIDPo machines recalculated tpm_aggregate matches with the signed tpm_aggregate, this means that the EUm and GIDPo smls are untampered, or vice versa.

6.3 GRID machine's distrusting/ trusting

Distrusting/ trusting the federated GRID attesting EUm and GIDPo machines is associated with each and every sml item by comparing their measurement values with the trusted measurement sml items. In the federated GRID model, each GIDP needs to have its own policies to handle: (1) fingerprint classification, and (2) fingerprints that are distrusted (or unidentified).

6.4 GRID machine's binaries privacy

In the suggested rma protocol for the federated GRID model, the end-user's (e.g., EUm) machine publishes its significant information to the GIDPo to form trust. In our approach, the privacy of EUm machine binaries is protected because all of the end-users of a particular domain are in a strong trust with the entity GIDP (home organisation).

6.5 GRID machine's patches updating

In the proposed solution, we assume a single domain contains a particular set of end-users (i.e., 70-200) so it is easy for a domain administrator or IT staff to update the patches easily. Also, it is the duty of a particular domain administrator to have a policy not to allow the list of measurements to grow further than practical limits. Also, a policy needs to be in place to update the domain end-users' machines' measurements at the GIDPo only when new programs are installed or old programs are removed.

6.6 GRID machine's transaction integrity

To access a federated GRID resource via a web-portal the EUm/ GIDPo machine's integrity is important during a complete transaction processing (e.g., before and after a web request/response) to a domain GIDPo (web-server).

6.7 Machine's key certification and bindings

In the proposed solution, we presuppose a domain administrator or IT staff is responsible to enable, activate, and own the TPM's of a domain end-user's and GIDPo's machines. A domain owner is also responsible for AIK generation, registration, and certification (i.e., if TTP (PCA) is present). The $AIK_{certificate}$ states the association of the AIK to an authentic TPM. The authentic TPM $AIK_{certificate}$ then can then be used in the TPMQuote. We also presuppose that the GIDPo (a challenger) has the legitimate and trusted certificates binding RSA AIKpub keys of all of a domain's all end-users' and GIDPo's machines TPM's. The AIKpub keys can be used by the GIDPo to verify the end-users' machines' TPM Quotes before the measurement list validation.

7 Conclusions and Future Work

For the purpose of this paper, the IMA based mutual integrity challenge protocol with the UN/PWD solution was suggested for the federated GRID model to validate/verify the mutual integrity of the EU_m and $GIDP_o$. The solution presented combines the basic AuthN mechanism, TPM, and the *rma* scheme. In the federated GRID model, the EU_m AuthN, the EU_m and $GIDP_O$ machines' mutual attestation, and the AuthR processes are executed in a sequence. The TC hardware security chip TPM combination with the *rma* scheme offers: (1) TC technology enhancing the GRID machines' security by using the cryptographic hardware chip (TPM), and (2) affixes trustworthiness to the GRID's ecosystem (i.e., the EU_m and the $GIDP_O$ machines' mutual integrity measurement, reporting, verification, and resource access). In our approach, trust is constructed between the GRID entities ($GIDP_O$ and GSP_n) via the federation concept and "behaviour trust". For example, a $GIDP_O$ EU_m machine is not allowed to use a protected resource if the EU_m/ $GIDP_O$ machines are not in a trustworthy state. Also, Shibboleth allows the $GIDP_O$ EU_m and attested machines' anonymity at the GSP_n by the attributes that do not include any machine binaries' identifying information. In this scheme, the *entity-GVO* may employ the notion of dynamic or variable attributes to let the GVO handover the "*twi*" or "role" attributes in a response to the GSP_n.

The federated GRID model verification, the protocols' (*rma* and lma) tests, and results' assessment are the future works.

8 References

[1] Frank Manion, William Weems, and James McNamee. "Federated Authentication". Biomedical Informatics for Cancer Research, Springer, USA, pp.18-25, 2010.

[2] Shashi Kiran, Patricia Lareau, and Steve Lloyd, "PKI basics - A Technical Perspective". PKI forum, Nov 2002.

[3] Trusted Computing Group (TCG). At: http://www.trustedcomputinggroup.org

[4] Bajikar, "TPM-based Security on Notebooks". Technical Report, Intel Corporation, 2002.

[5] Härtig. "Authenticated Booting, Remote Attestation, Sealed Memory aka "Trusted Computing". technische universität Dresden, summer semester, 2007.

[6] Gajasinghe, Introduction to SAML, Feb 2014.

[7] Wu and Periorellis. "Authorisation- Authentication using XACML and SAML". 2005.

[8] Ian Foster. "The Globus Toolkit for Grid Computing". International Symposium on Cluster Computing and Grid's, 2001.

[9] Yuri Demchenko. "Overview of Existing and Developing Systems for Authentication and Authorisation and Policy/role based Privilege Management". 2003.

[10] Shibboleth plus PERMIS integration. Sec.cs.kent.ac.uk/permis/integrationProject/ShibbolethDetails.html

[11] Von Welch. "Gridshib: Grid-Shibboleth Integration (Identity Federation and Grids)". eScience Security Workshop, UK, April 2005.

[12] XML (Extensible markup language), w3.org/XML.

[13] David Chadwick, Andrey Novikov, and Alexander Otenko. "GridShib and PERMIS Integration". Computer-wide Information System, Vol.23, No.4, pp. 297-308.

[14] Von Welch, Tom Barton, Kate Keahey, and Frank Siebenlist. "Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration". 4th Annual PKI R&D Workshop, 2005.

[15] David Chadwick. "The EC PERMIS Project" Salford, UK.

[16] Tom Barton, Jim Basney, Tim Freeman, Tom Scavo, Frank Siebenlist, Von Weleh, Rachana Ananthakrishnan, Bill Baker, Monte Goode, and Kate Keahey. "Identity Federation and Attribute based Authorisation through the Globus toolkit, Shibboleth, GridShib, and Myproxy". 5th Annual PKI R&D Workshop, 2006.

[17] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leenderi van Doorn. "Design and Implementation of a TCG-based Integrity Measurement Architecture". 13th USENIX Security Symposium, pp. 223-238, 2004.

[18] Ahmad –Reza Sadeghi, and Christian Stuble. "Property-based Attestation for Computing Platforms: Caring about Properties, not Mechanisms". Workshop on Network Security Paradigms, New York, USA, pp-67-77, 2004.

[19] Morgan, Scott Cantor, Steven Carmody, Waiter Hoehn, and Ken Klingenstein. "Federated Security: The Shibboleth Approach". EDUCAUSE Quarterly, Vol.27, No.4, 2004, pp.4-6.

[20] Rashmi Bhatia. "Grid Computing and Security Issues"; International Journal of Science and Research Publications, Vol.3, No. 8, 2013.

[21] Zubair Ahmad Khattak, Suziah Sulaiman, and Jamalul-lail Ab Manan. "A Study on Threat Model for Federated Identities in Federated Identity Management System". Information Technology International Symposium (ITSIM), Malaysia, Vol.2, pp. 628-623, 2010.

APPENDIX A

TABLE II. NOTATIONS USED IN THIS PAPER

Acronym	Meaning
EUm	the *m*th end-user
GSPn	the *n*th grid service provider
GVO	the grid virtual organization
GIDPo	the *o*th grid identity provider
PTR_{GSPn}	the GSPn protected target resource
TPM_{EUm}	the *EUm* trusted platform module
TPM_{GIDPo}	the *GIDPo* trusted platform module
$GIDPo_{IDp}$	the *GIDPo* identity IDp
$GSPn_{IDq}$	the *GSPn* identity IDq
RMA	the remote mutual attestation unit
AVC	the attestation validation unit
AC	the attestation caller unit
ACU	the attestation collector unit
AA/GAA	the attribute authority/ grid attribute authorithy
Ar	the attribute requester
AR	the attribute resolver
TwIR	the two way integrity resolver
AIK	the attestationidentitykey
PCR	the platformconfigurationregister
N1,N2	the nonce
v	the PCR value
vc	the virtual cookie
Sml	the store measurement log
prv & pub	the private and pub keys
SAML	the Security Assertion Markup Language
XACML	the Extensible Access Control Markup Language
AuthN	the authentication
AuthR	the authorization
UN/PWD	the username/password
PKI	the public key infrastructure
IDP	the identity provider
SP/RP	the service provider/ resource provider
VOs	the virtual organizations
ID	the identity
TC	the trusted computing
TPM	the trusted platform module
RA	the remote attestation
rma/RMA	the remote mutual attestation scheme
PBA	the property based attestation
IMA	the integrity measurement architecture
EK	the endorsement key
OASIS	the organization for the advancement of structure information standard
SMTP	the simple mail transfer protocol
HTTP	the hybertext transfer protocol
SOAP	the simple object access protocol
FTP	the file transfer protocol
PDP	the policy decision point
SSo	the single sign-on
SSTC	the security service technical committee
PCA	the policy combining algorithm
RCA	the rule combining algorithm
GTkit	the globus toolkit
PERMIS	the privilege and role management infrastructure standard validation
GridShib	the grid shibboleth
RBAC	the role based access control
FId&AM	the federated identity and access management
HTTPS	the hypertext transfer protocol (secure)
SSL/TLS	the secure socket layer/transport level security
lma	the local mutual attestation
AIK	the attestation identity key
TCG	the trusted computing group
PV	the privilege verification component
GIDP	the grid identity provider
GSP	the service provider
ABAC	the attribute base access control
PEP	the policy enforcement point
PIP	the policy information point
PAP	the policy access point
FId	the federated identity

Network Security Threats and Vulnerabilities

Manal Alshahrani , Haydar Teymourlouei
Department of Computer Science
Bowie State University,
Bowie, MD, USA

Abstract - *The transfer of confidential data over the Internet has become normality in the digital age with organizations and individuals using different digital platforms to share confidential information. This private information has become a target for hackers. With hackers targeting these networks, there has been a growing need to protect data, hardware, and software from vulnerabilities. A broad definition of network security can be constructed by defining its two components, security and networks. Two of the main focuses of this paper are to define network threats, such as phishing email, and to discuss some anti-phishing techniques. This research investigates various tools to identify different types of vulnerabilities and threats to the critical infrastructure and also identifies the network vulnerability and prevention methods for the network threats.*

Keywords: *network security, hackers, attack, vulnerabilities, threats*

1 Introduction

Security threats affecting networks are complex and pervasive in nature. To successfully protect a system from threats and vulnerability, it is essential to understand how security professionals assess and determine risks, the definitions of threats, exploitation, and vulnerability, and how security mechanisms are used. A threat may be demonstrated as intent to harm an asset or cause it to become unavailable. Identifying threats is an important but extremely complicated aspect of security management. Vulnerability, on the other hand, can be defined as flaws or weaknesses in system security procedures, design, implementation, or internal controls. Vulnerabilities can be accidentally triggered or intentionally exploited, resulting in security breaches. Security is a term used to describe different situations such as a situation without risk or sense of threat, prevention of risk, or a sense of confidence. A threat is an event that can take advantage of a vulnerability and cause a negative impact on the network. With the increase of universal electronic connectivity, threats such as eavesdropping, hackers, fraud, and viruses have grown exponentially. The fast growth of computer networks and systems has increased the need for individuals and organizations to store their information electronically or to use these systems for communication purposes. With more and more individuals and companies engaging in digital platforms, there has been a need to raise awareness about the importance of protecting data and resources, offering authentic messages and data, and protecting systems from network-based attacks.

Network security is not meant only for computers with significant data such as those used in businesses and offices [8]. Home users can also benefit from securing their networks. It is also important to note that not only broadband users or individuals with high-speed connection need to secure their networks. Another significant piece of information to have in mind is that the majority of computer systems, including corporate ones, have no immediate threat targeting their data; rather, compromised systems are used for practical purposes, for example, a launch of a DDOS attack in opposition to competing networks.

Furthermore, securing computer networks can be complicated. Historically, only qualified and experienced experts have been taxed with securing networks. However, as more people become concerned about potential security threats, there is need for more individuals who can understand the basics of the principles of network security and the network security world in general [9]. Different organizations and individuals are in need of appropriate security; therefore, the level of security varies from one organization to another. To be able to ensure better security for oneself and one's organization, it is crucial for network users to use the systematic approach, which includes analyzing, designing, implementing, and maintaining a desired network security system. The analysis phase requires a complete investigation of an entire network system, which is inclusive of both the hardware and software. This is an important phase because it helps to establish the level of vulnerability within the system and the requirements needed to ensure that the system is secure. Nowadays, all of the major browsers on Windows and Mac OS X are vulnerable to attack, so there is a significant increase in the use of HTML attachments to deliver malicious content. Typically, phishers will set up a fake login screen on a web page and then send spam emails with links to the site to as many people as possible in an attempt to trap them. For the attacker phishing, it is desirable to have a general method of spoofing any URL without relying on temporary exploit or clever domain registration. In this paper, we identify phishing attacks as a security problem resulting from utilization of Unicode on the Internet; demonstrate this potential and dangerous attack; and propose corresponding countermeasures to solve these kinds of problems.

. 2 Research Methodology

To achieve our goals, we will investigate following parameters. The comparative research of Linux versus Windows versus UNIX network security focuses on which operating system has the better security tools and is more secure. It also focuses on the different threats and

vulnerabilities that can affect each type of operating system, different types of security attacks, and available security countermeasure tools, techniques, and essential open source security tools. The results of these findings will be based on the simulation experiment.

2.1 Network Security Comparison

2.1.1 Linux operating system

With the increase in internet traffic, more and more transactions are taking place. These transactions are at risk since people with ill motives can target these transactions with the aim of damaging, stealing, intercepting, or altering data. Linux based systems are popular because they have robust and sophisticated security measures. Linux security tools can be broken into if they are poorly implemented into a system. Usually, attackers exploit existing problems within the system although the Linux community is quick at spotting these exploits and releasing immediate fixes.

The Linux server offers different kinds of facilities such as mail, WWW, and ftp which it handles via the system of ports. For example, port 21 controls ftp [2]. To be able to save on system resources and make the system administration less complex, many services configure file/etc/inetd.conf. This is the file that informs the system how to run each available service. Many Linux vendors turn on the different inetd.conf services by default; however, to ensure maximum security, they need to be off to prevent accidental damage.

Linux enables a user to choose which hosts to allow or deny. For instance, a user can allow logins from machines available at their own site, but not from the Internet. The files /etc/hosts.allow and /etc/hosts.deny list allowed hosts and services. The method of limiting or denying connections by checking the host provides a fundamental method for discouraging attacks although it is possible to fake host names on incoming connections. Data that is transferred over the Internet is also in danger because anyone with knowledge can gain access to the information using a method referred to as spoofing. Spoofing enables unauthorized individuals to inject fake data into a legitimate stream. The way Internet protocols interact enables such problems to occur. To tackle these difficulties, ssh was created.

Ssh is a stable, well-developed open source system that provides authentication and encryption on connections through the use of codes to protect data while it is in transit. The authentication process allows for the verification of a packet of data to ensure that the connection is valid. By using Linux, an individual is able to provide ssh level security for an individual's network use.

Linux has a comprehensive group of subsystems that enable a systems administrator to know what is going on with her or his system. All manner of log files are kept in the /var/log directory. Most of the basic services log information to /var/log/syslog and /var/log/messages about network users attempting to connect or successfully connecting to them.

Linux offers tools such as Ethereal which help to capture various types of packets over a period of time, revealing different types of information about packets. It is a critical tool used for monitoring the movement of packets and also for detecting traffic on a network segment. Tripwire is another intrusion and logging tool that takes a snapshot of important system files and records their signature within the database. After the initialization of the database, users can use Tripwire to monitor the integrity of the system. Another program is referred to as Snort, which offers information on the number of access attempts to which a machine has been subjected.

2.1.2 Windows Operating System

Windows is considered a secure operating system. It offers an improved and more secure computing experience, which is founded on user feedback from the Vista experience during which users requested for more intuitive and user-friendly security features.Windows was developed according to the Security Development Lifecycle (SDL) and it is built to be a secure computing environment. It contains key security features, including Data Execution Prevention (DEP), Kernel Patch Protection, Mandatory Integrity levels, and Address Space Layout Randomization which provide a strong foundation that guard against malicious attacks [4]. Windows has an enhanced User Account Control, primarily built to force software developers to engage in better programming practices; it is also perceived as a security feature that aids in enforcing least-privileged access and improves the total cost of ownership. The features allow organizations to deploy the operating system without granting administrator access to users. To improve security in a machine, experts recommend a two-factor authentication. Namely, this involves adding a second layer of protection on top of a password for improved security. Many computers, especially laptops, have built in biometric security systems in the form of a fingerprint scanner. Windows provides easier and more reliable support for integration between the fingerprint-scanning hardware and the operating system. Configuring and using a fingerprint reader with Windows for gaining access into the operating system and also for user authentication is more efficient and allows for up to ten finger scans.

Laptops can get stolen easily. If a user does not have effective security controls, unauthorized individuals can have access to sensitive information. Windows uses data protection technologies, including Active Directory Rights Management Services, Encrypting File System, and BitLocker that enable data encryption. From this information, it is evident that although the Windows operating system has better network security, Linux and UNIX have their own strengths, which can appeal to a user's preference.

3 Network Vulnerabilities

In a network, the first vulnerable assets are people because the majority of employees at a standard organization are not

particularly cautious about network security, and often, they will cause a security breach. Regardless of whether it was unintentional or intentional, this type of threat is called an insider attack. A threat that goes hand in hand with human error is social engineering, which involves taking advantage of gullible employees and gaining sensitive information or physical access. Typically, a social engineer will pretend to be a legitimate business representative, such as a member of a maintenance or repair crew, and ask to gain access to a server room or other restricted area. Other methods include false telephone calls, and email attachments and links to infected websites. It is the job of IT professionals and security management to prevent these kinds of attacks by providing security training to the staff and properly restricting access.

Although firewalls and antivirus technologies remain significant in blocking many external attacks and removing malware, there are some key internal threats that increase network vulnerability. This category includes a range of portable devices that can bypass firewalls and other network perimeter measures. These devices include, but are not limited to, the following: USB thumb drives and miscellaneous USB devices; optical media; laptops and netbooks; and smart phones and other digital devices (Manky, 2010).

USB or other mass storage devices have the capability to transfer and store large amounts of data; this means that they can be used to hold stolen data and at the same time, they can be easily disguised and transported out of the company. In addition to being portable for data theft, these devices can carry computer viruses and other forms of malware that can infect an endpoint computer. They can hide in many types of devices. For example, in 2008, Best Buy found a virus in an electronic picture frame that was used to hold photos (Manky, 2010). The malware can then wreak havoc on the system or install a backdoor which creates a network vulnerability that allows unauthorized users to access the network. A technique to mitigate a USB device problem is to disallow auto-run of these devices when connecting to a USB port. Other solutions come in the form of security policies implemented by management such as banning removable media from the workplace.

Another network vulnerability related to removable devices is portable end points and wireless access points. The portable end points are laptops and similar devices that can connect wirelessly or wire to a network. Once inside, these devices can scan the network for more vulnerable devices that they can infect. It is important to safeguard open RJ-45 ports that malicious users can connect to by turning off unused ports and restricting physical access. Laptops can also carry home sensitive company information, so it is prudent to use an encrypted file system to protect the data. Wireless access points are known to be vulnerable because of weak protocols, namely wireless encryption protocol (WEP). It is recommended to use stronger encryption protocols such as WPA2 with strong passwords and scheduled password changes to prevent brute force attacks.

3.1 Tripwire's SecureCheq

Tripwire's SecureCheq is a free network vulnerability scanner that has a simple user interface and scans for advanced Windows settings to check for vulnerabilities. Although limited to Windows systems, it can perform local scans of both desktops and servers. The categories of settings that it checks for are OS hardening, data protection, communication security, user account security, and logs and auditing. After a scan is complete, the results of each test for the settings are displayed in a summary report. In addition to the summary report, it provides a test report that holds individual test results, suggest methods of mediation, and references additional vulnerability information. The downside is that the results cannot be saved, which means it cannot be processed by an external script. However, the results can be printed and the OVAL XML file can be exported and saved.

The test under OS hardening includes the following: Windows Remote Desktop configured to allow only system administrator access; Windows Remote Desktop configured to always prompt for password; and an enabled safe DDL search mode. For the Windows Remote Desktop settings, allowed users and password prompts can be configured in local or group policy. These settings are necessary so that regular users cannot gain remote access into a server; another layer of security is provided when the password is entered manually. Safe DDL search mode can be enabled in group policy as well. This setting is important because it points DDL queries to the more restricted Windows system DDL files first instead of local DDL files which can be corrupted.

Under the data protection category, the settings focus on preventing anonymous access to Windows shares and disabling default guest accounts. These settings can be configured in a group policy object under security settings. Restricting anonymous access on shares is important, but a related topic is setting proper access controls on shares for users that do have access. This prevents users from deleting other's work by restricting them to their own files or folders. Windows guest accounts and other default accounts should be disabled because of they are often the target of hackers who seek to use them as backdoors into systems.

In the communication security section, the settings focus on enabling stronger encryption and disabling weaker encryption standards. The settings are also configured in group policy. The first step is making sure encryption is required for Windows network passwords, which prevents sending passwords in plain text over the network. It is also recommended to disable LM authentication in favor of stronger NTLMv2 authentication. The other settings cover encryption over remote sessions which protect data from eavesdroppers.

The user account security category focuses on Windows password and account lockout policy. These settings are located within group policy. It is recommended that Windows password complexity is enabled along with a minimum password length of at least eight (8) characters. This helps prevent brute force or dictionary attacks that attempt to guess a password. The account lockout policy should be configured

to a duration of at 15 least minutes, and the counter reset should be at least 15 minutes as well. This helps deter or slow down an attacker trying to guess a password or login.

The last category that the free version of SecureCheq tests for is logs and auditing. The majority of these settings are configured in group policy, and specific details are configured in the auditpol command line tool. It is recommended that the system and security log files' maximum size be large enough to hold the events but small enough increase system performance. Some of the recommended auditing settings to enable include the following: logging of executed applications; logging of credential validation; logging of successful and failed login attempts for domain and local accounts; and logging of successful system changes.

3.2 Angry IP Scanner

Angry IP Scanner is a network scanner that is open-source and can work on any platform. It is able to function on a Linux system as well as a Windows system and Mac OS X. There are many different features in Angry IP scanner. These features are good to have when scanning a network. One feature is the ability to set an IP range, which is very convenient if certain IP addresses should be scanned. As a result, time will not be wasted scanning unnecessary hosts. With this method, the administrator can enter the IP address to start from and the IP address to end the scan. The IP addresses can also be randomized. If the administrator is unaware of what IP addressed to scan, Angry IP scanner will be able to choose for them. The actual process of scanning is very quick. After the scan, the amount of time the scan took to run is displayed as well as the IP range scanned, the amount of hosts scanned, and the amount of hosts that are alive. Another feature that is useful is the identification of the hosts that are alive or dead. If the IP address has a blue circle beside it, the host is considered alive. If there is a red circle beside it, the hosts are not alive, and there is a dead ping. You can also right-click on a specific host, and some options such as Show Details, Rescan IP, Delete IP, Copy IP, Copy Details, and Open are shown. You are also able to copy the details and paste them into a report or save the results in a Word document.

All of the options are self-explanatory in what they do, but the "Open" option has different choices when you click it. These options include Edit Openers, Windows Shares, Web Browser, FTP, Telnet, Ping, Trace Route, Geo Locate, and E-mail Sample. Angry IP gives these options under "Open" because the IP address can be opened in those particular ways. If the IP address is opened in "Windows Shares," it will show all the shares associated with that IP address. If opened in "Web Browser," it will open the web UI where settings can be changed, etc. If opened in "FTP," it will show the FTP. The user also has the option to "Edit Openers," which allows the addition or deletion of options to one's preference. Also, penetration tests can be conducted.

3.3 Browsers Vulnerabilities

3.3.1 Simulated browsers attack

This simulation presents an example of phishing attack 'phishing email' to offer comprehensive information on phishing. We used simple method "Email / Spam" based on our project requirement in an Information Security and Privacy course. The assignment required students to investigate network attacks and list the tools that exciting with this attack to familiarize students with different types of attacks. The purpose of this study was to help students gain a better understand of how this attack works and how they can avoid falling victim to phishing attacks. Additionally, the project was designed to give students insight into how extensive a problem phishing is in information technology. Ultimately, the study enables us to present some phishing protection tips to safeguard the network and identity.

3.3.2 How does phishing work?

Phishing is the attempt to gain sensitive information such as usernames, passwords, and credit card details, in order to carry out identity theft using fraudulent e-mail messages that appear to come from legitimate businesses. Phishing is typically carried out by email spoofing [5] or instant messaging [6], and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one. Once the user visits the website, any information entered on the webpage will be collected by the phisher and may be used fraudulently.

Type of phishing attacks:

In this section, we give a brief overview of the different types of phishing attacks to familiarize readers with the various threats.

1. Phishing-Link Manipulation:

   Most methods of phishing use some form of automated trickery considered to make a link in an e-mail emerge to belong to the spoofed organization. It could use misspelled URLs or subdomains, tricks commonly used by phishers. Another common trick is to make the anchor text for a link appear to be a valid URL when the link actually goes to the phishers' site.

2. Filter avoidance:

   Phishers have used images in place of text to make it harder for anti-phishing filters to sense text commonly used in phishing e-mails

3. Phone phishing:

   An example of phone phising is a message that appears to come from a bank that instructs users to dial a phone number to resolve time-sensitive problems with their bank accounts. Once the owner (victim) provides his or her personal information, including account number and PIN over IP service, the phisher (hunter) will capture it and use it to the detriment of the owner.

4. Website fake:

   Some phishing scams use JavaScript instructions in order to alter the address bar. Mainly, they direct

the user to sign in at their bank or service's own web page, where everything from the web address to the security certificates appear exactly like the trusted organization's. Phishers started to make a new technique to a void anti-phishing website that examine website for phishing they begun to use Flash-based websites to keep look much like the real website, but conceal the text in a multimedia object.

According to a study by Gartner, 57 million US Internet users have identified the receipt of e-mail linked to phishing scams, and about 2 million of them are estimated to have been tricked into giving away sensitive information [7].

Figure1: Phishing Report In Various Activity.

3.3.3 Phishing Scenario

The approach used to simulate the problem was to create a phishing website and attempt to retrieve user's credentials. The testing environment comprised of creating an html file that contained a modified webpage with changed parameters, a php script that would record the victims' input and redirect them to the real login website once the login button is clicked, a log file that stored credentials in clear text, an email to send to the victim, and a php hosting website to host the malicious website. Once the victim receives the phishing email and clicks the malicious link, he or she will be directed to the phishing website. The layout of the phishing website will be an exact copy of the legitimate website, and, therefore, the victim would input their credentials. Upon clicking the login button, the php script would run, save the inputted credentials to the designated log file, and redirect the victim to the legitimate website. The log file is then examined and the victim's credentials are successfully retrieved.

3.3.4 Phishing web hosting

On a website where the users are supposed to enter/submit data (e.g., email, password), there is a piece of code in html called as action form.

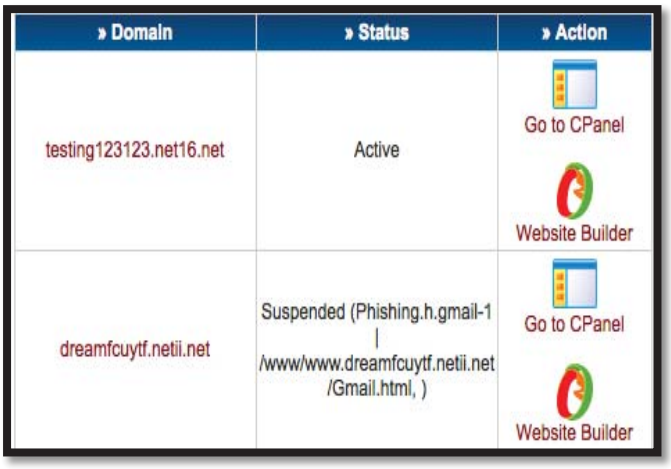


Figure 1: The Domain Website Builder

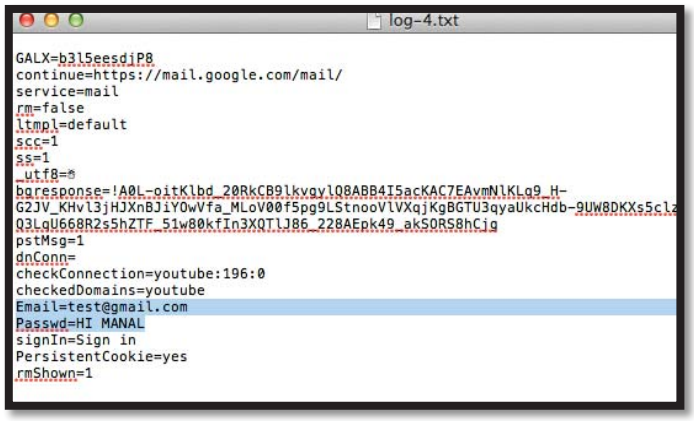


Figure 2: Log File Has the User Information

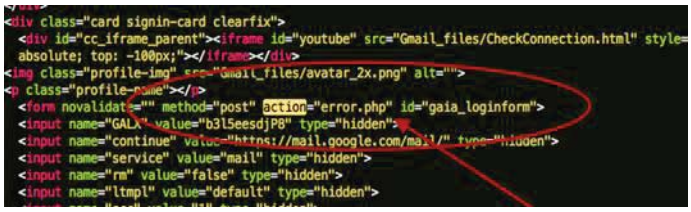


Figure 3: Place the Meta Tag to the Compromised Website

Figure 4: The Link To Fake Website

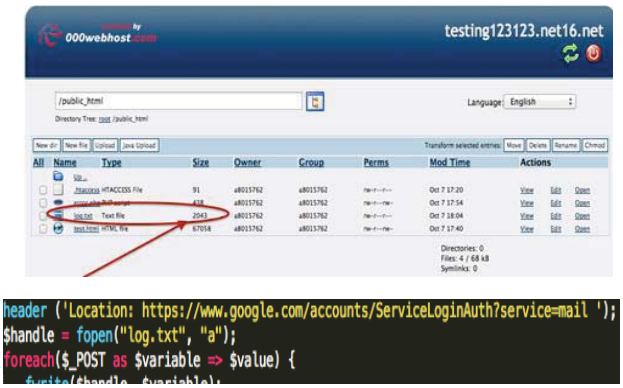


Figure 5: Code Capture Password

We captured how many times users success and log in spam phishing link where they are redirecting them, so we can have their information.

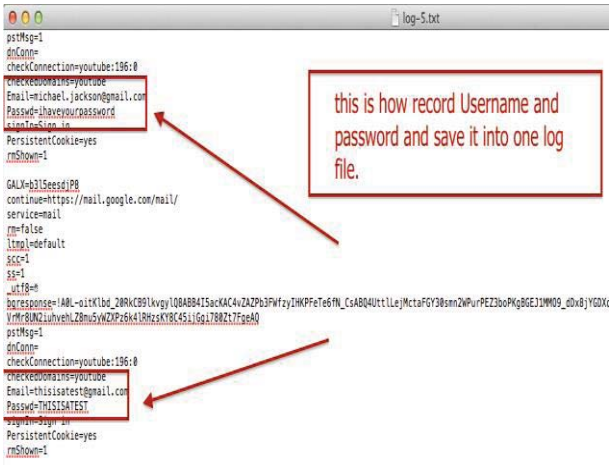


Figure 6: The Domain Website Shows the Number of Users Log In

Once the e-mail is opened by the user, the e-mail contents have to be sufficiently realistic to cause the recipient to follow the directions in the e-mail.

3.3.5 Phishing Prevention

Tips for phishing prevention incude the following: (1) Use dedicated systems for payment requests and approval processes. (2) Apply disabling email access on any system involved with payment processing. (3) Never trust alarm emails. (4) Be aware of attachments, and make sure not to open suspicious or strange emails, especially Word, Excel, PowerPoint, or PDF attachments, (5) Check the website you are visiting to ensure that it is secure; when you visit the page, examine the address bar; if the website you are visiting is on a secure server, it should start with "https://" ("s" for security) rather than the usual "http://"; most of phishing emails will direct the victim to pages where entries for financial or personal information are required. (6) Be wary of emails that request personal data, and never enter financial or personal information into kind of these pages. (7) Ensure that your software and firewalls that will enhance your defense against attackers are up-to-date; firewall protection prevents access to malicious files by blocking the attacks; antivirus software scans every file which comes through the Internet to your computer, which helps to prevent damage to your system.

4 Security Countermeasures Techniques and Tools

A security countermeasure is a device, procedure, technique or action that reduces a threat, attack, or vulnerability. Countermeasures against viruses includes having the latest software patches, service packs, and operating system; blocking irrelevant ports at the host or firewall; disabling unused services and protocols, and strengthening default configuration settings. To ensure your password is safe, users need to use strong passwords and audit login attempts to observe whether or not there have been hacking attempts.

4.1 Security Free Tools.

To be able to protect oneself from attackers and network vulnerability, there are free tools that can be used. These tools include, but are not limited to:

1. Wireshark: a multi-platform open-source network protocol analyzer that allows users to examine data from a live network or by capturing a file on a disk [10].
2. Metasploit: an elite open-source platform that enables users to build, test, and exploit code. This tool is perfect for exploitation research.
3. OpenVas: a tool that scans for vulnerability.

5 Recommendations

✓ For users to be able to protect themselves from phishing, it is important to have better email spam filters, two factor authentication, a site key, and an understanding of the factors that can enable and prevent phishing.

✓ The researchers would like to investigate potential methods of countering or stopping phishing attacks. Since the time window between the start and end of a phishing attack is likely to be limited to a matter of only hours or days and the source hosts are widely distributed, this is a difficult task.

✓ The researchers investigated how they can concentrate on collecting phishing emails received by end users. While this is a viable approach, capture occurs at the final stage in the incident lifecycle. An automated approach to capturing and responding to phishing attacks would be more desirable.

✓ If we hope to design web browsers, websites, and other tools to shield users from such attacks, we need to find a novel browser extension such as AntiPhish, that aims to protect users against spoofed website based phishing attacks.

6 Conclusion

From exploring network vulnerabilities and threats and by examining the different strengths of operating systems such as UNIX, Linux, and Windows with regards to security, it is evident that security is not a specific brand, product, firewall or operating system. Properly configured firewalls, antivirus updates, and passwords management strategies are good security practices; however, deficiencies in bad products can inhibit the results from these good practices. Therefore, it is essential for users to invest in software and hardware that are capable of withstanding common threats that might interfere, enable modification, fabrication, or facilitate interception of data. Our study suggests that a different approach is needed in the design of the security system rather than approaching the problem solely from a traditional cryptography-based security framework. In addition, most experts agree that anti-phishing education for end users needs to be implemented better. To resolve this matter, we recommend further study of development and innovative ways for combating anti-phishing attacks by finding better email spam filters, Two-Factor Authentication, or SiteKey.

8 References

[1] Ahmad, N., & Habib, M. K. (2010). Analysis of

Network Security Threats and Vulnerabilities by Development & Implementation of a Security Network Monitoring Solution.

[2] Chen, X., Mu, B., & Chen, Z. (2011). NetSecu: A collaborative network security platform

for in-network security. In Communications and Mobile Computing (CMC), 2011 Third International Conference on (pp. 59-64). IEEE.

[3] Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. Dependable and Secure Computing, IEEE Transactions on, 10(4), 198-211.

[4] Cole, E. (2011). Network security bible (Vol. 768). John Wiley & Sons.

[5] "Landing another blow against email phishing (Google Online Security Blog)". Retrieved June 21, 2012.

[6] Tan, Koontorm Center. "Phishing and Spamming via IM (SPIM)". Retrieved December 5, 2006.

[7] McDaniel, Robert. "Cyveillance Weekly Phishing Report - September 21, 2015." Cyveillance Blog The Cyber Intelligence Blog RSS. Cyveillance Blog - The Cyber Intelligence Blog, 21 Sept. 2015. Web. 18 Mar. 2016.

[8] Dave Dittrich, Network monitoring/Intrusion Detection Systems (IDS), University of Washington.

[9]Wright, Joe; Jim Harmening (2009) "15" Computer and Information Security Handbook Morgan Kaufmann Publications Elsevier Inc p. 257

[10] Sanders, Chris (May 23, 2007). "Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems". No Starch Press: 192. ISBN 1-59327-149-2.

[11]Ciampa, M. (2010, Jan. 29). Network Vulnerabilities and Attacks. Retrieved Sept. 20, 2015, from slideshare.net

[12] Manky, D. (2010, Nov. 8). Top 10 vulnerabilities inside the network. Retrieved Sept. 21, 2015, from networkworld.com

[13] *The Need for Vulnerability Assessment. (2013). Retrieved Sept. 20, 2015, from beyondtrust.com*

SOC as a Service : a user centric approach for Network Security Monitoring

A. Nicolas Grenèche[1], B. Mohamed Koné[2], and C. Christian Toinard[3]
[1]DSI / LIPN, CNRS, UMR 7030, University of Paris 13, Paris, France
[2]DSI, University of Paris 13, Paris, France
[3]INSA CVL, Bourges, France

Abstract— *Cloud infrastructures enable customers to build logical and isolated environments above shared physical ressources. This isolation heavily relies on system and network virtualization. A Security Operation Center (SOC) is a centralized unit that deals with security issues. SOC IT security specialists analyze network events to detect security violations. In the end, they advertise customers with detected threats. This paper proposes a novel approach where the network provider delegates the SOC management in a consistent manner to the different customers i.e. the tenants. Thus, the tenants share the same SOC that is controlled by the Cloud network provider. It gives a clear separation of privileges where the Cloud network provider controls the SOC while a tenant controls his Cloud Infrastructure and the corresponding delegated SOC. Thus, the proposed SOC service supports autonomous cloud infrastructures where the provider only is responsible for the network infrastructure and the different tenants are responsible for their cloud infrastructure. This service enables multi-tenants to consult their private and consistent network events i.e. consolidated events spawned by various network security monitoring tools (traffic analyzer, NIDS etc.). The monitoring tools are dynamically setup and administrated by the cloud network provider. Thus, the tenant's effort is reduced since the delegated SOC is dynamically configured according to the deployed serviced limiting thus the work of the tenant to the analysis of his threats.*

1. Introduction

This paper introduces the notion of Security Operation Center (SOC) as a Service : SOCaaS. SOCaaS enables tenants to access Network Security Monitoring (NSM) events without configuring network security tools for their Cloud infrastructure. Thus, NSM events are formally defined and consolidated through data coming from various network security monitoring tools. Moreover, each tenant manages its own Cloud infrastructure while controlling a dedicated but consistent private SOC. Thus, SOCaaS is a user centric approach that fits with dynamic Cloud infrastructures where the tenants can customize and use a consistent and secured SOC with a low effort.

The first contribution is to define a model for NSM events. This model is closely related to the Open System Interconnection model of the International Organization for Standardization [1]. Our model permits to fit each tool in a NSM event format that is compliant with the OSI model. Thus, multiple network security tools can provide different information related to the same NSM event. These heterogeneous informations are consolidated in a consistent NSM format. The tenant will access to that consolidated NSM event.

The second contribution is to perform a dynamic management of the network services authorizing thus independence between a customer and the cloud network provider. Network services are setup freely by customers in their own Cloud infrastructure. Those services appear and disappear dynamically on the cloud infrastructure without advertising the cloud provider. Our approach provides an inventory of network services running on the tenant's Cloud infrastructure. This inventory enables to 1) configure the security monitoring tools and 2) adjust the access policy to consistent NSM events i.e. related to the updated view of the services. Thus, the cloud provider provides a dynamic management of the delegated SOC.

The third contribution is to define an access control model dealing with the NSM events. A NSM event is basically an interaction between two network end-points : a service and a client. For this purpose, a formal language enables to model the delegated SOC policy. This policy his shared between the network provider and the corresponding tenant, First, the provider prevents illegal accesses to other tenants whatever is the shared policy. Second, the shared policy allows the tenant to adjust the monitoring of the discovered services. Thus, security specialists at the tenant side have a fine control over the monitoring. Moreover, the network provider has precise directives to automatically adjust the monitoring enforcement according to the requested policy.

The last contribution is the implementation of such a SOCaaS within the network infrastructure of the Paris 13 University. It provides a real and large experimentation of the proposed approach for sharing the SOC between the different tenants of the Paris 13 University.

2. State of the art

SOC's IT security analysts [2] monitor the network interactions of the whole infrastructure. They watch for malicious activities in the network interactions. The SOC can be distributed to facilitate Computer Network Defense between

ISBN: 1-60132-445-6, CSREA Press ©

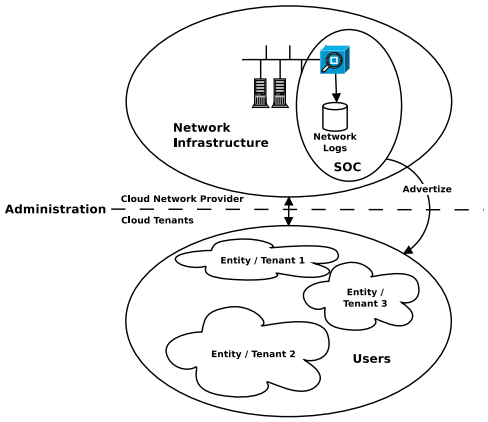


Fig. 1
A CLASSICAL SOC

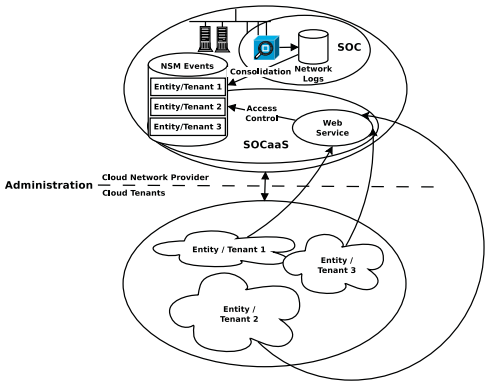


Fig. 2
THE SOCAAS ARCHITECTURE

multiple subordinate distinct SOCs. However, the coordinating SOC often has limited authority over its subordinate SOCs. The coordinating SOC can not help or even provide automation and can not guarantee consistency for the different entities/tenants associated to these SOCs. Thus, the classical approach does not fit with dynamic and autonomous Cloud infrastructures since the tenants need automation and security guaranty for their dedicated SOCs.

The same network interaction can generate data from different monitoring tools. NSM specifies several kinds of network interaction data [6]. This paper relies on session, transport and alert data.

- Core data contains addresses of endpoints and OSI layer transport protocol ;
- Session data contains endpoints and metrics of the interaction, e.g., number of packets or size of data ;
- Transport data are applicative request and response, e.g., HTTP get request and it's return code ; Transport data include the application data e.g. the login and password for accessing to a webmail service.
- Alerts are raised by NIDS on suspicious network activities.

There are four other levels of data that are out of scope of this paper : comprehensive capture, statistics of capture, metadata (e.g, geographical location of interaction endpoints) and content data (i.e, extraction on files from network flows). Some visualisation tools implement a pivot method to automatically switch from a level of data to another [7]. However, the NSM approach does not tell how to consolidate the information coming from different network monitoring tools into a consolidated NSM event i.e. a NSM event including the different informations/alerts coming from the various tools.

Authors are proposing NSM approaches for the Cloud. They either consider collaboration of different network tools [3], cloud management of the security policy [4] or virtualization of the security services [5]. However, current solutions do not consider a SOCaaS approach where the network provider offers a Cloud service with automation and autonomous configuration of the network monitoring. Such a SOCaaS service must offer a consistent view of the monitored security tools without the tenants having to deal with the configuration of security monitoring and must present a consolidation of the NSM events. Finally, the SOCaaS approach must enable a tenant to control his consolidated NSM events but prevent him to access the events from the other tenants.

SOCaaS addresses the corresponding limitations :

- automatic inventory of the tenants network services;
- automatic configuration and deployment of various network security monitoring tools;
- computation of consolidated NSM events using elementary data from the various NSM tools;
- consistent view of the NSM events regarding dynamic network services;
- protection of the NSM events against the other tenants;
- shared monitoring policy between the network provider and each tenant;
- user centric approach where the tenant finely controls the monitoring through his own NSM policy;

3. The SOCaaS architecture

In order to adress those different limitations, a global approach is proposed.

Figure 1 presents the classical SOC approach. The Cloud network provider manages the network infrastructure and a global SOC. The Cloud provider advertises the tenants responsible of the different Cloud infrastructures with the corresponding threats. Sometimes the SOC is distributed over subordinate SOCs but in this case each entity/tenant is responsible for the deployment and administration of his SOC. Such a situation does not fit with distributed Cloud infrastructures where a network provider wants to supports his customers with a SOC facility.

Figure 2 presents the architecture corresponding to the proposed SOCaaS approach. The Cloud network provider manages the network infrastructure. A global but dynamic SOC 1) discovers the services, 2) computes the consolidated

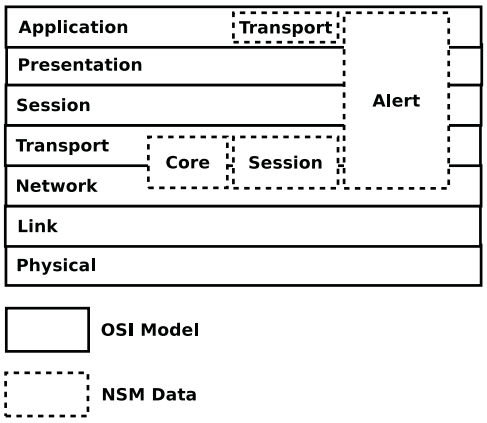


Fig. 3
NSM INTEGRATION WITH OSI

NSM events, and 3) updates the view of the NSM events according to the changes. Moreover, an advanced web service allows each tenant to access his NSM events through a secure but user centric control. That web services presents several advantages. First, the web service prevents a tenant from accessing over tenants events. Second, it authorizes the tenant to define its own monitoring policy. Such a policy can allow the tenant to select the services he wants to control, preventing thus useless monitoring of some services. Moreover, that tenant policy authorizes a fine grain configuration of the monitored services without having to deal with the underlaying network monitoring. Thus, the SOCaaS approach offers a clear separation of privileges between the Cloud network provider and the tenants of the different Cloud infrastructures. It shares the management of a dedicated SOC between the network provider and each tenant while reducing the work of the both partners.

4. Consolidated NSM events

First, let us explain how all the data levels of the NSM approach fit with OSI model. Figure 3 shows correspondances between the NSM levels of data and related OSI layers. Security analysts manually switch from a level of data to another. For example, if an alert is raised, analysts have to check transport data to retrieve applicative protocol used on this interaction.

As one can see, the NSM model does not reinvent the OSI model. But it takes a different view of the 7 levels of the OSI model. Indeed, NSM sees the transport as a mean for an application to communicate its data. In NSM the major problem is to deal with application threats. So, the OSI application is viewed as 1) a transport part (e.g. HTTP) that deals with the communication channels whereas the remaining part is 2) the true communication of the application data e.g. login and password. But, as one can see the alerts are considered for all the real software layers of the model OSI. Indeed, the alerts cover the network to application layers of the OSI model.

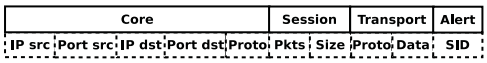


Fig. 4
MINIMAL NSM EVENT

A NSM event is a combination of every piece of information supplied by various network monitoring tools. The minimal core information supplied by every monitoring tool are : source and destination address / port and OSI layer 4 transport protocol. Core information are used as a key to combine all these data according the NSM classification in a NSM event. An example of minimal NSM event is given in figure 4. In this figure, the NSM event is composed of the four classes of NSM data : core, session, transport and alert. Each class is composed of attributes. Core attributes are source and destination address / port and OSI transport protocol. Session attributes are the number of packets and the size of the communication. Transport attributes are the applicative protocol and related data. Alert attribute is the SID (Signature IDentifier) of the detected threat.

Monitoring tools provide NSM data in several forms : files, databases etc. These are data sources for NSM events. Agents are automatically deployed and poll on the data sources to extract attributes of NSM classes. Every extraction must contains core attributes. Agents use a key derived from core attributes to combine the other attributes into a consolidated NSM event.

An exemple is given in figure 5, a computer from the local network sends an HTTP get request to a remote web server. The request goes through different network analysis tools : a NIDS, a flow collector and a protocol analyzer. All these tools provide various information about the NSM event, respectively : alert, session and transport. Each agent polls on a data source to get some attributes of the NSM event. Agent 3 gets the HTTP request and the answer from the server. The GET request raises an alert from the NIDS. The domain is referenced as a malware domain in a signature database. Agent 2 extracts the SID of that alert. At the end of the communication the flow collector logs the communication. Agent 1 retrieves the session information for the communication. These data are combined thanks to the key derived from core attributes to generate the consolidated NSM event related to this communication. The sequel will explain that the access fo the NSM events is governed through clearance based on a subject-access-object set of rules. But, let us now explain how the list of monitored network extremities is dynamically updated in order to give a consistent view of the core data.

5. Dynamic management of the network services

The same security threat can have a different impact considering its source or destination. If the threat targets a

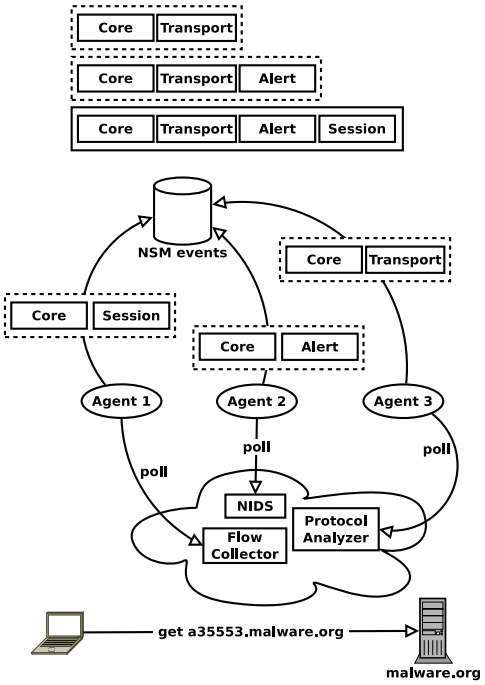


Fig. 5
CONSOLIDATING A NSM EVENT

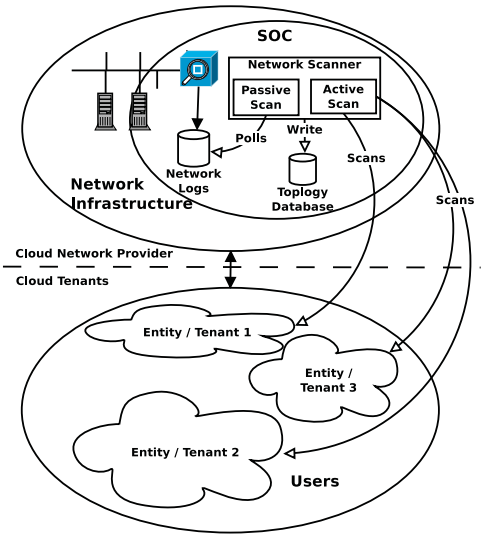


Fig. 6
NETWORK ACTIVE / PASSIVE SCAN

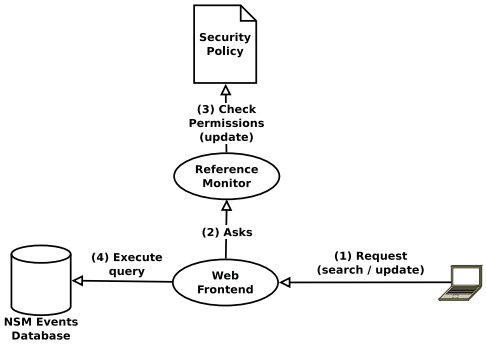


Fig. 7
AUTHORIZATION PROCESS

network service belonging to the tenant environment, it is an intrusion attempt. If the threat comes from a host belonging to the tenant environment, it is an extrusion. In this case, the internal host has been compromised. The compromission is a consequence of a successful intrusion.

Cloud network provider supplies routing to tenants and maintains NSM tools suite. As a consequence, he is able to 1) know which tenant is attached to which network(s), 2) access network traffic of each tenant. Tenant installs network services without notifying the cloud network provider. Service topology evolves in an unpredictable way for the cloud network provider. Cloud network provider must dynamically detect services. We use active and passive methods to detect services. Figure 6 shows how the dynamic management of services interacts with the cloud network provider infrastructure.

The active method consists in using a network scanner to detect services on networks attached to each tenant. The main asset of this method is that results are very accurate and comprehensive (e.g. version of the service, operating system etc.). The drawback is that services on non-standard or filtered ports will be missed. To mitigate this drawback, we also use a passive method.

The passive method benefits from the access to the network traffic. NSM session data ara analyzed to detect successful connection to services running on top of cloud network provider's IP scopes. Detected services are added in the topology database regardless it is filtered or binded on a non-stantard port. These two methods are combined to obtain a precise topology for dynamic network services.

This topology is stored in a database in order to be reused for the access control to NSM events.

6. Access control

The access control to NSM event is based on the Type Enforcement (TE) model [8]. In a TE policy, access is governed through clearance based on a subject-access-object set of rules.

The security policy defines who can access to which NSM events. In the same tenant, many identities can access the SOCaaS. These identities are subjects and NSM events are objects. The access control must meet two objectives 1) a tenant cannot access to another tenant NSM events, 2) identities belonging to a tenant can have different permissions. The cloud network provider knows tenant's IP scope, so the first objective is a mandatory rule of the access control : a subject of a tenant cannot access to NSM events that are not related with one of his networks. This control is performed on source / destination attributes of the core part

of the NSM event. We defined a formal language to meet the second objective. This language defines a set of access rules for an identity on NSM events based on a set of criterias :

access *subject* **on** *net* **direction** *dir* **attributes** *attrs [with grant]*
subject : email adress of the user
net : network address | all
dir : inbound | outbound | both
attrs : session &| transport[=*protos*] &| alert &| all &| none
protos : protos, proto
proto : http | dns | ssh | ...

Subject is the physical person that takes part in NSM in the tenant.

Net is a network allocated by the cloud network provider to the tenant. Keyword *all* stands for all networks allocated by the cloud network provider to the tenant (according to objective 1).

Dir is the direction of the interaction. *Inbound* means that the interaction comes from a host outside of a tenant's network and reaches a tenant network service. *Outboud* means that the interaction comes from a host inside a tenant's network to a remote network service. *Both* means *Inbound* and *Outbound*.

The differenciation between *Inbound* and *Outbound* is possible thanks to the dynamic management of the network services. Source and destination of interaction are checked against the topology database to deduce the direction of the NSM event.

Attrs is a list of attributes accessible by the subject. These attributes are *session*, *transport* and *alert*. If *none* is specified, user will only have access to the core part of the NSM event. The *transport* attribute can take in argument a list of protocols to select only NSM event matching specified applicative protocol(s).

The *with grant* option defines the *subject* as administrator for the network *net*. This *subject* can modify the security policy for the network *net*.

The authorization is performed as shown in figure 7. The client sends a request to the web frontal. There are two types of request : search and update. Search requests enable to access NSM events. Update requests perform a modification of the security policy. The frontend asks the reference monitor if the request is allowed regarding the submitter identity. The reference monitor guarantees that the request conforms with the security policy. If the request is allowed, the reference monitor updates the policy (update request) or grant an access to the database of NSM events (search request).

For example, we consider a tenant using the subnet X. In this tenant there are three persons in the IT staff : Admin, Tech and Web. Admin is a network administrator that 1) can access all the NSM events and 2) can modify the security policy. Tech deals with user's applications. He wants to watch for supicious activities coming from those applications. Web wants to collect statistics on the web services. Web services use the address Y which belongs to the subnet X. Let us show the policy defined at the tenant side :

access *admin* **on** *all* **direction** *both* **attributes** *all* **with grant**
access *tech* **on** *all* **direction** *outbound* **attributes** *alert*
access *web* **on** *Y* **direction** *both* \
attributes *session,transport=http*

Let us explain the tenant policy. Admin has an unlimited access to every NSM events. Tech can only access to NSM events, originated from the tenant's network, that contain an alert. He will obtain a list of potentially compromised hosts. Web can access to the NSM events of machine *Y* related with a HTTP activity.

As one can see the tenant only has to define a policy regarding his monitoring requirements. While the network provider only enforces the requested tenant policy. Thus, a clear separation of concerns is supported where the tenant defines the monitoring requirements and the network provider satisfies those requirements within dynamic services. The tenant does not have to set-up a new policy for monitoring new services. Indeed, the policy supports dynamic services. However, the tenant can finely configure the monitoring without dealing with the management of the underlaying NSM tools.

7. Experimentation

The experimentation provides a practical implementation of the proposed approach for sharing the SOC between the different tenants of the Paris 13 University. The number of network services and bandwidth has drastically increased last years making impossible for the CISO to analyze every event. We performed an inventory of services listening on University network. Results are obtained from both active and passive scanners (ANS and PNS).

The ANS discovered 1025 hosts with 4526 TCP services. The median of TCP services per host is 3 and the average is 5. UDP port scanning has been disabled for a faster probing. The PNS discovered 4 hosts (that have not been seen by the ANS) and 13832 TCP services. The large number of TCP services is explained by the fact that PNS adds a service for each TCP connection established with a computer that belongs to a subnet of the dynamic environment. In the dynamic environment, there are some hosts that act as media stream providers receiving a lot of incoming TCP connections. A median of 2 verifies this hypothesis. The PNS has also detected 327 UDP services with a median of 1 and an average of 3 (the median gives a more realist picture of the rate of services per host).

Concerning the amount of data to be analyzed, an extraction from Argus log files shows a global throughput around 600MB/s on the monitoring interface. Inbound traffic (destined to tenants' network services) is around 15 MB/s (4% of the total throughput). An instance of Snort is running on this interface.Inbound traffic raises 12,6% of the whole alerts.

Majority of these alerts are applicative request nomalies : 85,2%. Then, transport anomalies raise 12,3%. The remaining 2,5% are malicious payloads (mainly SQL injections and remote exploits). Outbound traffic is around 585 MB/s (96% of the total throughput) and raises 87,4% of the whole alerts. Applicative and layer anomalies represent 94,1% of extrustion alerts The remaining 5,9% are malwares, communications to blacklisted addresses and port-scans.

The experimentation aims at providing an access to each University entity IT administrator (our tenants) to network events. CISO keeps on configuring and maintaining network probes (NIDS, network traffic analyzers etc.). However analysis is distributed to tenants. The IT department of the University acts as an IaaS provider for tenants. We develop a layer between tenants and network events that provides 1) consolidation of network events, 2) access control on consolidated network events to share them between tenants in a secure way and 3) a web service to access network events. These three components enable to provide a SOC as a Service for University tenants. The SOCaaS implementation relies on three programs :

- A network scanner that performs active and passive scans to maintain the topology database of tenants ;
- A NSM event collector that polls on the NSM tools to generate NSM events ;
- A web service is available to the tenants thanks to a dedicated reference monitor that controls the accesses to the NSM events.

Our SOCaaS relies on several NSM tools for network data source : Argus, Bro and Snort. Argus is used to collect metrics of the network usage on the monitoring interface. Bro gives information about both session and transport attributes. Snort adds alert information to the NSM event. This analysis suite runs on a off-the-shelves Linux server . This server is connected to a mirroring port: all the traffic is duplicated to this server interface. The NSM tools suite has been bound to this mirroring interface.

The network scanner written in Python takes the list of tenants' networks as input. The active scanner loops on this list to scan each host using the Python-nmap API. Results are stored in a MySQL database. This database is the topology database. The passive scan polls on Bro conn.log (network connection logs file in plain text) and looks for connections to a network service running on top of an address belonging to the cloud network infrastructure. Detected services are added to the topology database.

The NSM event collector is presented in figure 8. Bro generates several log files in plain text. Each file is related to an applicative protocol. The agent spawns a thread for each file. Each thread polls on its file to get new connection logs. When a new connection is logged in the file, the agent retrieves information and puts them in the NSM event format. Then, the agent writes information in an intermediary Redis database.

This database contains NSM events that are not consolidated yet. NSM tools are asynchronous. Data composing a

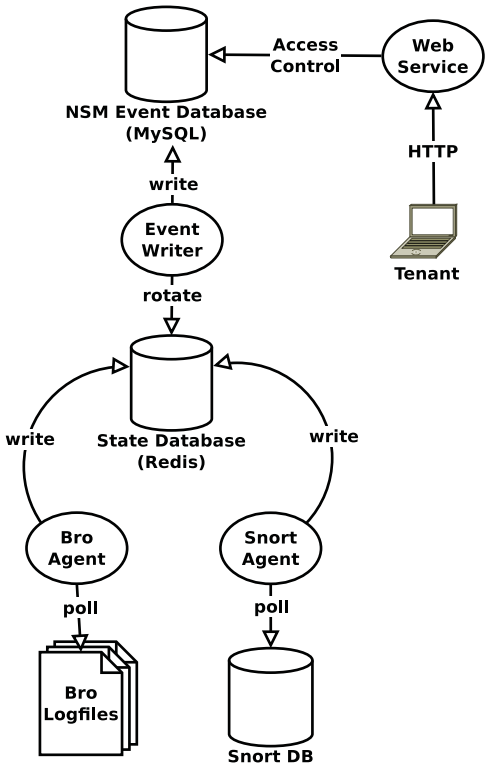


Fig. 8
IMPLEMENTATION

NSM event are not available at the same time. Redis database is a volatile state database. NSM agents get core attributes and a subset of the NSM event. NSM events are consolidated thanks to the key derived from core attributes 3. An agent stores the NSM event in the Redis database. We used Redis because it's a all in memory database (fast and volatile) optimized for key lookup operations (consolidation of NSM events through core attributes). The event writer performs a rotation of the NSM Redis events into a MySQL database. The rotation puts every NSM event older than 5 minutes (arbitrary value) in the MySQL database.

The web service handles a client request for the MySQL NSM event database and checks that the request satisfies the security policy. This web service is implemented following architecture introduced in 7. The web frontal is written in PHP, the reference monitor is a Python daemon and the security policy is a plain text XML file.

8. Conclusion

This paper proposed a model that has been implemented on a real network. The process can be improved for detection aspects. First, some tenants may have high needs regarding confidentiality. They have to get rid of the NSM performed by the cloud network provider as their traffic must not be analyzed by another organisation. A tenant may also want to run its own NSM suite. SOCaaS can easily meet such a confi-

dentiality needing. To this (to do so, to realise this), SOCaaS can reuse the topology database to automatically configure the tenant's NSM suite running in the safe infrastructure of the tenant. Indeed, the Cloud infrastructure belongs to the tenant. The second major enhancement deals with the design of an autonomous web service for accessing the NSM events. This can easily be supported through a REST API authorizing a dedicated client to access the NSM events in a safe manner.

The SOCaas architecture can also be reused for protection. The tenant has access to its own network events. He can accurately decide whether an event is malicious or not. If a malicious activity is detected, we can enhance the security policy to enable the tenant to activate a set of pre-generated blocking rules for the targeted service.

Acknowledgment

We would like to thank the IT department of the Paris 13 University for its cooperation. Especially Mr Abdechchafiq, head of the network team, for providing us with an access to network traffic and Mr Chervet, IT vice-president, for funding hardware for this project.

References

[1] J. Day, *The (Un)Revised OSI Reference Model*, SIGCOMM Comput. Commun. Rev. Vol 25, n5, pp 35–55, ACM, New York, USA, issn 0146-4833, Oktober, 1995.

[2] C. Zimmerman, Ten Strategies of a World-Class Cybersecurity Operations Center, isbn 978-0-692-24310-7, Mitre Corporation, 2014.

[3] Zhen Chen, Fuye Han, Junwei Cao, Xin Jiang, and Shuo Chen, Cloud Computing-Based Forensic Analysis for Collaborative Network Security Management System, TSINGHUA SCIENCE AND TECHNOLOGY, ISSN 1007-0214, pp40-50, Volume 18, Number 1, February 2013

[4] Bobelin, Laurent and Bousquet, Aline and Briffaut, Jérémy and Caron, Eddy and Couturier, Jean-François and Lefray, Arnaud and Rouzaud-Cornabas, Jonathan and Toinard, Christian, An Advanced Security-Aware Cloud Architecture, The 2014 International Conference on High Performance Computing & Simulation, Bologne, Italy, IEEE, Jul, 2014.

[5] Li, Jianxin and Li, Bo and Wo, Tianyu and Hu, Chunming and Huai, Jinpeng and Liu, Lu and Lam, K. P., CyberGuarder: A Virtualization Security Assurance Architecture for Green Cloud Computing, Future Gener. Comput. Syst., vol 28, number 2, pp 379–390, Elsevier Science Publishers B. V., February, 2012

[6] Bejtlich, Richard, Practice of Network Security Monitoring, No Starch Press, July, 2013

[7] Phan, D., Gerth, J., Lee, M., Paepcke, A., and Winograd, T. (2008). Visual analysis of network flow data with timelines and event plots. In VizSEC 2007 (pp. 85-99). Springer Berlin Heidelberg.

[8] L. Badger et al. Practical Domain and Type Enforcement for UNIX. Proceedings of IEEE Symposium on Security and Privacy, May 1995.

SESSION

SPECIAL TRACK: END-TO-END SECURITY AND CYBERSECURITY; FROM THE HARDWARE TO APPLICATION

Chair(s)

Prof. Tiziana Margaria

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

SEcube™: an Open Security Platform - General Approach and Strategies

Antonio VARRIALE[1], Giorgio DI NATALE[2], Paolo PRINETTO[3], Bernhard STEFFEN[4], Tiziana MARGARIA[5]

[1] Blu5 Labs Ltd, Blu5 Group, Ta Xbiex, Malta – av@blu5labs.eu
[2] LIRMM, CNRS, Montpellier, France – giorgio.dinatale@lirmm.fr
[3] CINI Cyber Security National Lab & Politecnico di Torino, Torino, Italy – paolo.prinetto@polito.it
[4] Chair of Programming Systems, TU Dortmund, Dortmund, Germany – bernhard.steffen@tu-do.de
[5] University of Limerick and Lero, The Irish Software Research Centre, Limerick, Ireland – tiziana.margaria@lero.ie

Abstract *The SEcube™ (Secure Environment cube) platform presented in this special track is an open source security-oriented hardware and software platform constructed with ease of integration and service-orientation in mind. Its hardware component is a SoC platform: a single-chip design embedding three main cores: a highly powerful processor, a Common Criteria certified smartcard, and a flexible FPGA. The software components include several libraries of ready-to use components that provide developers with different entry levels to adoption. The software is modular, and available as API or as services in an advanced model driven design environment. This way, security experts can avail of the open source character, and verify, change, or write from scratch the entire system, starting from the elementary low-level blocks, but at the same time we support also developers who use the predefined primitives and can experience the SEcube™ as a high-security black box. This paper explains its aims and architecture, while the other papers detail the unique aspects of the overall platform.*

Keywords — Security, Hardware Platform, Software Development Environment, Model Driven Design, Open-Source.

1 Introduction

Security is a key concern to any mission and business critical service and application. Wherever there is data or proprietary or personal content there is a need of mechanisms and provisions in place to ensure adequate users' privacy, prevent and handle the commercial and legal issues related to security threats, and to safeguard the business stakeholders.

The implementation of a suitable security layer usually requires special skills in several disciplines, including mathematics and cryptography. When the security target is provided as a combination of hardware and software solutions the system complexity increases and further skills are required (i.e. electronics, physics, informatics) to manage the integration and prevent all the possible attacks. At the same time the security may have a big impact on pre-existing systems and solutions, most of the times forcing the final users to change their habits. All these aspects make the security a very complex topic from the development stage to the final usage.

In this paper and track we describe the architecture and the main characteristics of the SEcube™ (Secure Environment cube) platform, that is new in being consequently open and service-oriented. It is easy to integrate and capable of hiding significant complexity behind a set of simple and high-level services, that are accessible as APIs, the lowest level being the hardware itself. Also the hardware is open: its description and technical details are completely available, including the hardware netlist, the package, examples of schematics in order to plug the chip into possible boards. The complete software stack is open source as well, including libraries of services at different abstraction level offering palettes of pre-defined services, and the service APIs, up to the model driven development environment itself, that that enables the easy creation of new (secure) applications and products.

While the open platform concept is popular in the software domain (think of Linux, Java, and its spread in software development communities, like in bioinformatics), it is not yet commonly adopted for hardware, especially for security purposes, where the solutions are usually provided as a black box sometimes coming with a certification and always closed to any kind of disclosure or customizations.

There are a few security-oriented open platforms available on the market. Some of them are focused on the evaluation of the system robustness against external physical attacks (e.g., Side-Channel attacks, power crypto-analysis, etc.), such as the Sasebo board [1] and the ChipWhisperer [2].

ISBN: 1-60132-445-6, CSREA Press ©

Other platforms based on ARM processors, like Juno ARM Development Platform [3] and the open source USB device provided by InversePath [4], allow creating general purpose software applications, including security-oriented solutions. Nevertheless, they are based on application processors and there are not specific security elements to be fully controlled or customized by the developers.

Finally, there are single chips realized as a combination of one FPGA and one CPU, like Zynq proposed by Xilinx [5] or Excalibur based on Altera technology [6]. Nevertheless, in both the cases the platforms are more suitable at prototyping stage, since they are not cost-efficient, and a specialized security element, like a smart card, is still missing.

This paper describes in detail the SEcube™ hardware (Section 2) and software (Section 3) architecture, gives a brief introduction to the DIME environment that provides the eXtreme Model Driven Development design and verification capability (section 4), and concludes in Section 5 with an introduction to the other papers of this track. They provide in-depth descriptions of the security primitives and protocols, the design environment, several case studies, a first glimpse of how to deal in this overall context with property verification and enforcement.

2 The SEcube™ Hardware Architecture

As shown in Fig. 1, the hardware device is a single chip, which embeds three hardware components: a powerful CPU, a flexible FPGA and an EAL5+ certified smart card. The software library is developed as a free and open-source SDK, provided with user documentation.

Internally, the SEcube™ device is a multi-module chip integrated in a 9mm x 9mm BGA package. As shown in Fig. 2, it is a heterogeneous platform consisting of three elements.

The first element is a high-performance ARM Cortex M4 RISC CPU produced by ST Microelectronics [7]. It provides the following features:

- 2 MBytes of Flash memory
- 256 KBytes of SRAM
- 32 bit parallelism
- operating at frequency of 180 MHz
- dedicated FPU (Floating Point unit)
- Internal TRNG (True Random Number Generator)
- Hardware Crypto Accelerator
- Low power consumption

This CPU has been selected among many ARM based micro-controllers, since it offers several features that make it suitable for high-performance and security-oriented solutions. For example, it supports the Cortex CMSIS implementation that provides, among the others, the CMSIS-DSP libraries: a collection with over 60 DSP functions for various data types. The CMSIS-DSP library allows developers to implement complex, real time operations using the embedded harware floating point unit.

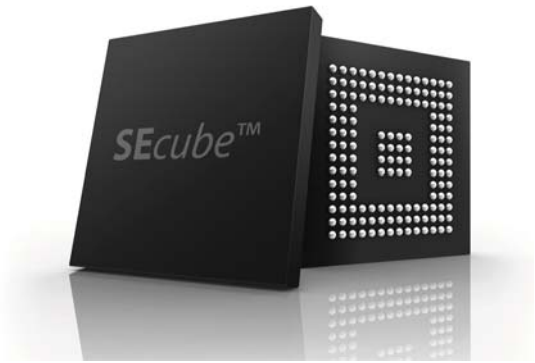


Fig. 1 The ***SEcube™ Chip***

In addition, the CPU provides several peripherals such as SPI, UART, USB2.0 and SD/MMC, which ease the hardware integration in diverse devices. For example, a secure USB device can be easily realized using the USB2.0 and the SD card interfaces, respectively.

On the security side, a TRNG (True Random Noise Generator) embedded unit, hardware mechanisms like MPU (Memory Protection Unit), and privileged execution modes allow implementing the security strategies required by a certified secure controller (e.g., privileged memory areas, key generation, etc.).

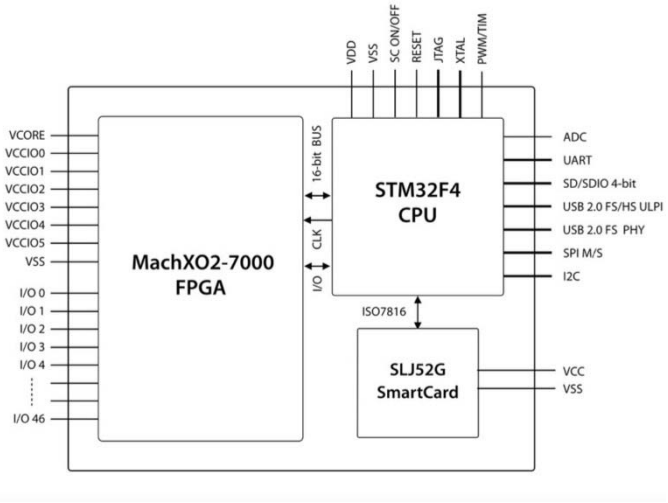


Fig. 2 ***SEcube Hardware Architecture***

For programming, debug and testing operations, the CPU provides a standard JTAG interface that can be permanently disabled once the development cycle is over, protecting all the sensitive information through a physical hardware lock.

The FPGA element, a Lattice MachXO2-7000 device [8], is based on a fast, non-volatile logic array providing the following main features:

- 7000 LUTs
- 240 Kbits embedded block RAM
- 256 Kbits user flash memory
- Ultra low-power device.

The FPGA exposes 47 general purpose I/O which may be used as a 32-bit external bus able to transfer data at 3.2 Gb/s. Inside the chip, it is linked to the CPU through a 16-bit internal BUS, able to reach a data transfer rate of 1.4 Gb/s. A CPU-FPGA clock line is also provided in order to simplify the clock domains synchronization.

In order to limit the number of pins and the BGA package size, the FPGA JTAG is connected just to the embedded CPU, which manages both the debug and the programming operations. As a positive side effect, the FPGA configuration can be implemented by means of customized, high-security techniques. For example, the programming stream can be encrypted and signed through dedicated algorithms. The CPU and/or the smartcard elements can decrypt and verify it before being injected in the FPGA.

The third component inside the SEcube™ device is an EAL5+ certified smartcard [9], based on a secure chip produced by Infineon that provides the following features:

- ISO7816 interface
- JavaCard Platform, Global Platform 2.2
- 128 KB Flash
- EC, ECDH up to 521 bit (HW accelerator)
- RSA up 4096 bit (HW accelerator)
- AES128/192/256 (HW accelerator).

As shown in Fig. 2, the CPU is connected to the embedded smartcard through a standard ISO7816 interface. The smartcard does not expose any interface outside the SEcube™ chip. This architectural decision provides high-grade and certified security functionalities behind a simpler and easy-to- use application interface.

Combined together, the above three components, allow to build a heterogeneous computing architecture and create the foundations to build a very flexible open source security platform.

2 The SEcube™ Software Libraries

The software side of the platform library consists of a multi-level, open source, collection of libraries available as an SDK [10, 11], together with a verification-oriented model driven design environment (currently DIME [12]). The libraries, especially if in conjunction with this design environment, allow developers who are not willing or able to produce the security primitives and protocols themselves to exploit the ready functions provided (as services or API) within the SEcube™ platform and experience the platform as a high-security black box. Conversely, security experts in the security domain can enjoy the openness and good documentation to verify, change or rewrite the pre-existing software starting from basic low-level blocks or even redefine entirely the whole system.

Leveraging the platform thought, we intend to create and nurture over time a community for developers at the different levels of security competence and in different application domains. This will ease the project, knowledge, and resource sharing and provide the collectivity of members with specialized support tailored to their needs.

From the architectural point of view, the software is divided in two main parts, depending on where physically the code runs: the Device-side relates to the SEcube™ based hardware device (e.g. USB secure token), while the Host-side relates to the appliance hosting the device (e.g. Laptop).

In this scenario the SEcube™ hardware device acts as a powerful coprocessor which provides a secure and fully controlled execution environment. All the functionalities implemented in the SEcube™ are thus exported to the host system through an open source secure RCP (Remote Call Procedure) protocol which is encapsulated in the SDK.

Before describing all the functional details (section 4), a general device and host side overview is given here.

2.1 Device-Side Software

The device side, software provides the libraries of basic functionalities that are executed on the embedded microprocessor. According to their purposes, the libraries cover three layers:

Functional *Layer0*. Closest to the device, **software drivers** offer basic functionalities to manage and access internal peripheral (e.g. TRNG, internal Flash memory, timers, etc), external devices connected to the CPU, i.e., FPGA and Smart Card, and the external communication interfaces as well (i.e. USB, UART, SPI and GPIOs). This level, in particular, is entitled to discover possible SEcube™ devices connected to the host system and create a bidirectional communication channel. These functionalities are exposed through a simple set of APIs:

- `L0_get_dev_list`, which is in charge of discovering connected SEcube based devices, and
- `L0_tx_rx`, which implements a low level send/receive operation on the channel.

Functional *Layer1*. At the intermediate level, the **core functions** constitute the basic primitives for implementing secure applications. This layer provides basic cryptographic algorithms and various utilities, like functions for power management.

In addition, it is used to secure the communication channel after a successful login, which can be easily implemented as a multi-factor authentication thanks to the three hardware elements inside the chip and the password provided by the user.

The Layer1 exposes several functions to manage both the device provisioning and the user/admin operational processes:

- pin and primary keys initialization (e.g., `L1_factory_init` and `L1_initLogin`)
- login (e.g. `L1_loginAdmin`, `L1_loginUser` and `L1_changePinUser`)
- logout (e.g. `L1_logoutAdmin` and `L1_logoutUser`)
- information retrieval (e.g., `L1_readDSN`, `L1_getAlgorithms`, and `L1_getKeyList`)
- encryption/decryption/signature verification (e.g., `L1_crypto_init`, `L1_crypto_setIV`, `L1_crypto_update` and `L1_crypto_finit`)
- key management (e.g., `L1_injectKey` and `L1_deleteKey`).

Functional *Layer2*. At a higher level, the **security abstraction layer** allows developers to create secure software and services for the applications (e.g., secure file system [10]), avoiding the need to understand in detail the low-level hardware and security mechanisms.

When the security target does not require a hardware implementation, the SEcube device can be virtualized by a software library which provides all the layers above running on the host.

2.2 Host-Side Software

On the host side, the software is tailored for existing devices (e.g., laptops or Desktop PC) that see the SEcube hardware as an external peripheral and use it for the specific functionalities offered, like cryptographic hardware acceleration. For this communication, the SEcube device is seen as a closed black box providing utility services. The host starts the service request by sending the related command and the optional data packets, through a proper interface, according to a custom protocol.

Also the host side code is open source. It is designed to be both scalable, e.g., for dealing with multiple devices, and portable on different operating systems, thus limiting the usage of and isolating platform-dependent modules.

The host-side software runs on top of the host operating system, and I structured in two main levels.

Layer0 implements the basic functionalities to communicate with the SEcube, including (a) sending/receiving command and data packets from/to the device, (b) segmentation of raw data streams into protocol-compliant packets, (c) functions implementing standard cryptographic algorithm and (d) low-level error management functions. Moreover, commodities functions are also provided, such as low-level data manipulation in dealing with possible endianness mismatches between the host side and the SEcube™ embedded CPUs.

Host-side software relies directly on the Operating System calls and it supports many OSs and platforms, including Microsoft Windows, Unix-like environments, and MacOS. To improve portability, OS-dependent sub-modules (e.g., communication interface, file system, etc.) are easily identifiable.

Layer1 is built over the Layer0 and provides a higher abstraction library, like multi-factor login, secure communication channel, cryptographic algorithms and key management. As we see in Fig. 3, currently, the Layer 1 library includes 16 functions, of which 6 concern login/logout (called `log` in the picture, which is their abstract type) 4 the key management (`key`), 4 the cryptography (`crypt`) plus 2 utility functions (`utils`). These functions can be combined to implement more complex security mechanisms at higher level, as detailed in [12,] [13], and [14].

The Layer1 also allows developers to manage several SEcube™ devices at the same time, providing dedicated operation control flows (one command/response session per communication channel), which allow encoding and decoding commands for the individual SEcube™ target.

As shown in [10] and [11] and sketched in the next section, the Layer 1 services are the basic building blocks for developers to create complex and tailored security primitives. As shown in [12] and [14], design of secure solutions based on these services eases the understanding and reduces the time to market drastically. The key to this speed and increased confidence in the correctness and security of an application is a model-driven design that includes these ready but customizable security services and protocols already at design time, on the models of any application. How the DIME environment supports this is explained in the next section.

3 Modeling Environment

DIME (the DyWA Integrated Modeling Environment) is an integrated solution for rigorous model-driven development of sophisticated and high assurance web applications. They are modeled in a simplicity-driven fashion that focuses on describing *what* application is sought (descriptive), instead of *how* the application is realized (prescriptive). Further design goals are agility and security as well as quality assurance. It is a consequent refinement of the realization of jABC4 (Java Application Building Center 4, [15]) for process modeling and DyWA (Dynamic Web Application [16]) for domain modeling and data persistence. The application workflow models in DIME are graph models: nodes are called SIBs, and represent executable functionalities, whose labeled outgoing edges, called branches, lead to the logically next appropriate SIB to execute.

In the spirit of its predecessors, DIME empowers prototype-driven application development following OTA (One Thing Approach [17]) and XMDD (Extreme Model-Driven Design [18]) by putting Subject Matter Experts (potentially non-programmers) in the core of the development process. Hence, different aspects of an application are described with the respective most adequate form of model. All these models are interdependently connected, collectively

shaping the One Thing model in a very formal yet easy to understand way. This is supported to the extent that the application can be one-click-generated to a running product.

The models created with DIME are transformed into code in a generation step where the complete target application is assembled according to the model's control flow and to the code of the elementary blocks, called SIBs. The target of this product generation is the DyWA framework, which constitutes the actual runtime environment, supports the deployment phase, and manages data persistence. However, the runtime platform, programming language, and frameworks are a matter of the corresponding (full) code generator, and may be changed without touching the models.

Consistent model-driven design together with the generative, service-oriented product assembly provide users with early prototyping of executable web applications as well as explicit support for product evolution, this due to the agile nature of version management for data management and persistency in DyWA. Altogether, the approach has the potential to tremendously push development cycles in an agile but consistent manner.

For model design, a family of Graphical Domain-specific Languages (GDSL) is tailored to express specific aspect of typical web applications:

- **Data models** cover the design of domain models based on common concepts like classes, attributes, and uni- or bi-directional relations between elements.
- **GUI models** specify the structure of (re-usable components of) web pages and the data binding in data sensitive user interface components.
- Different **Process models** types span the core business logic, data retrieval (search queries) as well as dynamic access control (security guards, particularly interesting for this platform).

Relations between these aspects are modeled by cross-referencing the models, this way creating hierarchical model structures.

Fig. 3 ***L0, L1, and SFS SIB palettes for the SEcube***

Such models are constituted by connected basic model components called SIBs (Service Independent Building Blocks) that either encapsulate other existing models or link to implementations, in form of code, or some form of API calls, e.g. services, RPC, other local or remote libraries. SIBs are the units of model re-use, and are well suited to represent library elements. For instance, the SEcube Layer0 and Layer1 API collections are seen in DIME's **Diagram Editor** as SIB palettes. Fig. 3 shows this for the Layer0 and Layer1 services, as well as for the Data at rest palette (Secure File System) of services described in [10], with a graphical representation of the SIBs within each layer. Each SIB in the L0 and L1 palettes corresponds to one of the functions discussed in the previous Section, with the L0 and SFS system SIBs spelled out and the L1 SIBs labeled according to their abstract types in DIME.

The inner logic flows of complex SIBs, once designed, are process models, represented as SLGs in DIME. The collection of available process models and the creation of cross-references are found in the **Models View**. The **Data View** lists data types and type attributes in a structured fashion. The **Properties View** deals with attributes and parameters. The **Model Validation View** manages the syntactic and semantic checks that provide guidance for the user and ensure correctness wr.t the current set of properties, as discussed in more depth in [14].

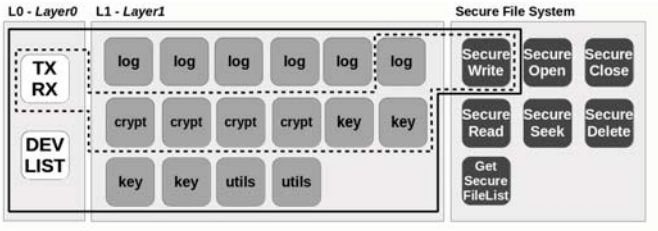


Fig. 4 ***Uses(x)*** *relation of the* ***SecureWrite*** *SIB of the Secure File System (SFS) palette*

Fig. 4 shows the subsets of L0 and L1 palettes used to implement a Secure Write operation within the Secure File System palette of the SDK described in [10].
As frequent for higher level functions, the internal behaviors of the Secure Write SIB are context dependent, and use different sets of functionalities in different contexts. This information and the ability to analyze and visualize it easily are important for any design decision that propagates across functionalities. For instance, when performing an impact analysis of changes in the underlying platform libraries. the Uses(x) relation is useful: similar to a call graph, or to a reachability analysis at the SIB level, Uses(x) for a given SIB returns the set of SIBs that occur in its Service Logic.

- The solid line encloses the SIBs used when the device is first inserted, **at connection time**. In this case, the logic flow internal to the Secure Write operation foresees that the device needs to be found using the L0 `get_DEV_LIST` SIB, then the L1 login procedures must be executed, followed by cryptographic and key handling functions, using also the utility functions. The communication happens all the time via the L0 transmit/receive functionality `TX_RX`
- For further SecureWrite operations carried out **after the first** one, the login has already taken place and the chip

has already the correct key set. In this case the logic internally executed is simpler, and uses only the SIBs enclosed in the dotted line. In this case, they are mostly cryptographic functions, and of course the L0 transmit/receive functionality `TX_RX`.

Already this simple example shows how the inner workflows of otherwise quite elementary operations can become orchestrations in the secure case, and how it may be useful to be able to guarantee properties of those flows, especially if context plays a role.

Throughout the development life-cycle, DIME design-time well-formedness and consistency, enabling one-click-generation and direct deployment of a sound and secure web application at any time. Continuous change and evolution are supported by means of iterative model modification, re-generation and re-deployment.

4 This Track

Addressing the path to End-to-end Security and cyber security, from the hardware to application, this special track acknowledges that security and cybersecurity are an increasing concern for all the actors in the IT and societal space. One of the limiting factors to integrated security is the lack of coordination between the different layers of security that individually cover the layers of the IT stack. Today, hardware, operating system, middle-ware, virtualization layer, and the many layers and components that constitute a user-facing application, including data and persistency, and the communication over networks, are still developed largely independently, with little inherent integration. Data at rest, data in motion, communication access and governance, as well as system evolution (e.g., through updates) are managed individually, too often under the responsibility of different actors, different technologies, different products and vendors. In this context, concerns of holistic security are growing, and call for a better end to end integration and communication across the different layers.

This paper illustrated how the partnership of different actors (an SME vendor in security with leading edge research institutions in hardware and software) has managed to create the SEcube™, an open source platform that integrates all these layers. The research and technology contributions concern the model driven and flexible integration and customization of security capabilities, features, and assets, rooted in the hardware device, and used and preserved in the software layers.

Other 4 papers of this track provide in-depth descriptions of the security primitives and protocols [10, 11], the design environment [12,14], several case studies in different application domains and technology spaces, ranging from web applications [12] to computer vision and robotics [14], and a first glimpse of how to deal with property verification and enforcement [14] in this overall platform .

Additionally, paper [19] concerns the Italian National Cyber Security Framework, a methodology that aims to offer to the organizations a volunteer approach to cope with awareness, management, and reduction of the cyber security risk. The Framework approach is deeply tied to the risk analysis rather than to technical standards. It is a generalization of the US NIST Framework for Improving Critical Infrastructure Cybersecurity and it has been realized in alignment with the NIST guidelines.

Acknowledgement

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

[1] Toppan Ltd, “Side-channel Attack Standard Evaluation Board: SASEBO”, http://www.toptdc.com/en/product/sasebo/, [Online; accessed 19-May-2016].

[2] C. O’Flynn, and D. C. Zhizhang. "ChipWhisperer: An open-source platform for hardware embedded security research." In: *Constructive Side-Channel Analysis and Secure Design*. Springer International Publishing, 2014. 243-260.

[3] Arm LTD, “Juno ARM Development Platform”, http://www.arm.com/products/tools/development-boards/versatile-express/juno-arm-development-platform.php, [Online; acc. 19 5.2016].

[4] Inverse Path Srl, “USB Armory”, https://inversepath.com/usbarmory.html, [Online; ac. 19-May-2016].

[5] L. H. Crockett, R. A. Elliot, M. A, Enderwitz, and R. W. Stewart. The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Ac. Media, 2014.

[6] Altera Corportation, “Excalibur Devices”, https://www.altera.com/products/general/devices/arm/arm-index.html, [Online; accessed 19-May-2016].

[7] ST Microelectronics, "STM32F4 Series Data Sheet - DocID022152 Rev 7", March 2016

[8] Lattice Semiconductor, “MachXO2™ Family Data Sheet – DS1035 – v3.2”, May 2016

[9] Infineon, "Infineon Chip Card & Security ICs Portfolio", October 2015

[10] A. Varriale, P. Prinetto, A. Carelli, and P. Trotta, “SEcube™: Data at rest & data in motion protection,” Proc. Int. Conf. on Security and Management (SAM), part of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[11] G. Di Natale, A. Carelli, P. Trotta, and T. Margaria, “Model driven design of crypto primitives and processes,” Proc. Int. Conf. on Security and Management (SAM), part of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[12] S. Boßelmann, J. Neubauer, S. Naujokat, and B. Steffen, “Model driven design of secure high assurance systems: an introduction to the open platform from the user perspective,” Proc. Int. Conf. on Security and Management (SAM), part of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[13] A. Varriale, E. I. Vatajelu, G. Di Natale, P. Prinetto, P. Trotta, and T. Margaria, “SEcube™: An Open-Source Security Platform in a Single SoC,” Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 2016 11th IEEE Int Conf. on, April 2016.

[14] G. Airò Farulla, M. Indaco, A. Legay, and T. Margaria, “Model driven design of secure properties for vision-based applications: A case study,”

Proc. Int. Conf. on Security and Management (SAM), part of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press

[15] B. Steffen, T. Margaria, R. Nagel, S. Jörges, and C. Kubczak, "Model-driven development with the jABC," In: *HVC 2006*, Haifa, Israel. Vol. 4383. LNCS. Springer, 2007, pp. 92–108.

[16] J. Neubauer, M. Frohme, B. Steffen, and T. Margaria, "Prototype-Driven Development of Web Applications with DyWA," In: *Proc. of 6th ISoLA*. LNCS 8802. Springer, 2014, pp. 56–72.

[17] T. Margaria, and B. Steffen. "Business Process Modelling in the jABC: The One-Thing-Approach," In: *Handbook of Research on Business Process Modeling*. IGI Global, 2009.

[18] T. Margaria, B. Steffen. "Service- Orientation: Conquering Complexity with XMDD". In: *Conquering Complexity*. Springer, 2012, pp.217–236.

[19] R. Baldoni, and L. Montanari, "End2End CyberSecurity, based on a strong Public-Private Partnership," Proc. of the Int. Conf. on Security and Management (SAM), part of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

SEcube™: Data at Rest and Data in Motion Protection

Antonio VARRIALE [1], **Paolo PRINETTO** [2], **Alberto CARELLI** [2], **Pascal TROTTA** [3]
[1]Blu5 Labs Ltd, Blu5 Group, Ta Xbiex, Malta
[2]Cyber Security National Lab, CINI & Politecnico di Torino, Italy
[3]Lero (The Irish Software Research Center), University of Limerick, Limerick, Ireland

Abstract – *Current trends for ubiquitous data usage have made information security as a mandatory component of any system. The availability of suitable levels of protection for data is required to secure any kind of content throughout its lifecycle and independently from the media, which allows the data to be used. In this paper we present a methodology to provide data protection through a simple and effective security abstraction layer based on the SEcube™ (Secure Environment cube) single chip, a new security-oriented open hardware and software platform. After analyzing the most critical information states, we introduce a set of easy-to-use APIs that provide an open-source, multi-paradigm security layer, suitable to protect both data at rest and data in motion. Being the SEcube™ made up of three hardware elements (a highly powerful processor, a Common Criteria certified smartcard and a flexible FPGA), all the functions are implemented and executed in a fully controlled secure environment. All the complexities related to key management and algorithms are handled within the secure environment, leaving the developers free to focus on the final applications and services.*

Keywords: Security, Data Protection, Hardware Platforms, Open-Source

1 Introduction

Information and data exposure during its entire lifecycle is growing continuously. The constant increase of connectivity, bandwidth, and mobility allows hackers and malicious users from inside and outside organizations to steal and monetize valuable information such as medical records, intellectual properties, bank transactions, and national secrets.

Security is critical, but comes at a cost. Such cost is not to be seen just as a money matter, but as the effort required to integrate security into suitable vertical solutions and the additional time and effort spent by the users to learn and implement the new processes. In this paper we propose a solution aimed to reduce these three aspects to a minimum. The acquisition cost is lowered by resorting to the SEcube™ integrated platform; the integration cost is drastically reduced by a low-impact deployment of a layered security technology, and the complexity of security processes in daily tasks is conceived by an abstraction level which guarantees that user's habits remain unchanged.

Information lifecycle is conveyed through various technologies, each having one (or more) dedicated security solution(s). Some examples are HTTPS for web surfing, S/MIME for e-mailing, VPN for private networking, and many encryption software solutions for HDD storage.

In such a heterogeneous universe of technologies, this paper introduces a new methodology. More abstractly, we distinguish information as being, at any time, either *at rest* or *in motion* and we provide for each a solution that is independent from the underlying technologies: SE*file* protects data at rest and SE*link* protects data in motion.

These solutions, based on the SE (Secure Environment) technology, are implemented according to a common strategy that eases the above-mentioned costs [2].

In order to minimize the actual cost of the product, a multi-paradigm approach is provided. The clients can tailor the solution according to their needs, from a full standard and software-only implementation to a fully customized hardware implementation, from open-source to commercial IPs.

Organizations can adopt the SE based solutions on top of all the pre-existing data storage, management and manipulation systems, such as clouds, e-mails and web based services. SE-based implementations are portable, independent of the Operating Systems and multi-level. How to do this in the software and hardware development platform is detailed in [3] and [4].

Finally, the cryptography complexity is concealed by common abstractions like groups, scopes and policies, in such a way that even the concept of encryption key is invisible [4, 5]. This approach allows users to enjoy security with no impact on their current habits.

The following sections will explain the main features of these solutions.

2 Data at rest and data in motion

Before Internet and the wideband mobile connectivity diffusion, the information lifecycle was much more controlled. Nowadays, as shown in Fig. 1, every piece of data may run complex and unpredictable paths from its creation to its final archiving or destruction.

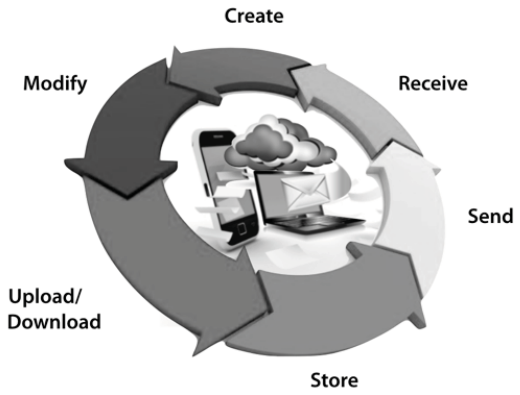


Fig. 1 – Information life cycle

In this scenario several kinds of contents (e.g., documents, messages, and voice/video streams) may require different protection techniques according to their state. For example, cryptography was created to protect communications defined as the transfer of sensitive information between two or more separate entities. This is the case of *data in motion*. However, when information is static, as for stored documents, we talk about *data at rest*.

Cryptography alone is not enough to guarantee a proper data protection, especially when the system weakness may vary according to the information state. It is important to identify a scalable and easy to use methodology able to protect both data at rest and data in motion. In the sequel we introduce such a methodology based on the SEcube™ security platform.

3 SEcube™ multi-device, multi-level, multi-paradigm libraries

The SEcube™ is an open source security-oriented platform in a single chip package produced by Blu5 Group. As described in [1] and [2], the SEcube™ chip provides several communication interfaces, which allow to embed it in any kind of device. In order to reduce the time to market, Blu5 Group also provides ready-to-use devices such as the USE*cube*, a powerful secure USB device which embeds the SEcube™ chip. Even starting from the development board (Fig. 2), OEMs and integrators can easily create diverse form-factor devices according to the final operational environment (e.g. USB tokens for PC/Laptop usage, PCI express boards for servers usage, etc.).

Fig. 2 – SEcube™ development board

On the software side, the SEcube™ platform comes with multi-level, hierarchical libraries, which provide three main levels of APIs:

- Level-0, Communication functionalities (e.g., driver-less USB communication channel) explained in [1]
- Level-1, Security common functionalities (e.g., login, logout, key inject/remove, basic cryptographic functions, etc.) explained in [1]
- Level-2, Service level functionalities (e.g., secure file system functions, negotiation) explained in the next sections.

Due to the open source nature of the project, cryptographic experts are free to modify the software implementations at any level, creating new libraries to their taste. A shown in Fig. 3, each functionality can be implemented in several paradigms according to the required security, performances, and compliance targets.

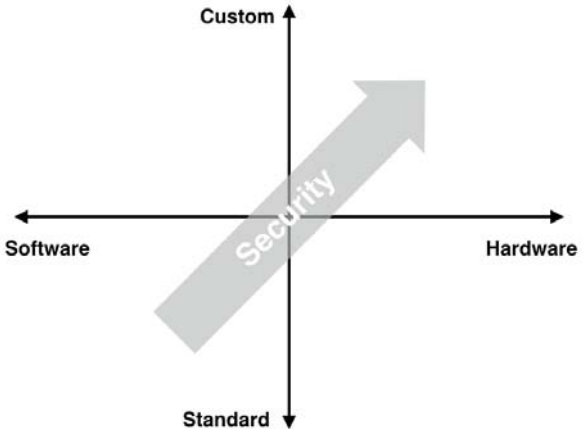


Fig. 3 – Multi-paradigm implementation diagram

For example, the basic cryptographic functions can be easily implemented as a firmware running in the SEcube™ embedded CPU. Nevertheless, an implementation based on the SEcube™ embedded FPGA can be considered when the performances in terms of speed are crucial. Alternatively, the use of the SEcube™ embedded smart card becomes mandatory when the final solution has to achieve specific certifications and compliances.

A software implementation is sometimes enough to guarantee the required security comfort. In this case a full software implementation of the SEcube™ (virtual SEcube™) can be used instead of the chip. In any case, whatever the final implementation is, the SEcube™ libraries are written in ANSI C in order to maximize their portability. Possible Operating System dependencies are isolated in a few modules and the whole code can be wrapped (e.g., JNI, PHP, JavaScript, etc.) to fit any development environment.

Providing a multi-device, multi-level, multi-paradigm approach, the SEcube™ platform is a flexible candidate to deploy security solutions in several scenarios, perfectly matching the final requirements even in complex and heterogeneous systems.

4 SEcube™ easy keys management

A security system becomes appealing when both developers and users are not aware of its complexity. For this purpose, the SEcube™ platform is based on concepts like *closed communication groups* and *security policies* other than keys and cryptographic parameters.

In both data at rest and data in motion based applications and services (e.g., local storage, cloud, email, messaging and voice calls) the sensitive information should be accessed and managed by a specific group of users, only. In the simplest case, the group is made up of one user (e.g., myself, for personal purposes). In other cases, many people can be involved (e.g., file transfer and chat rooms).

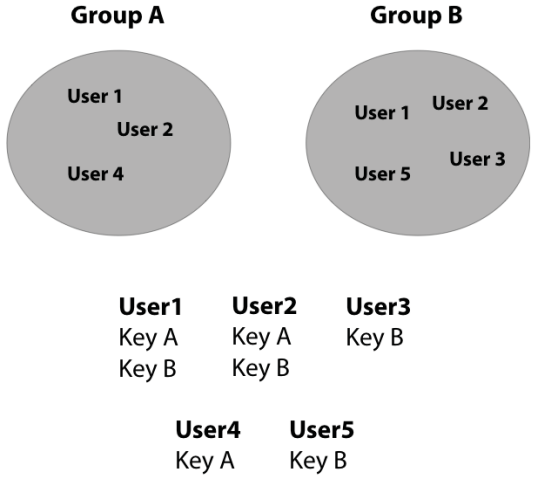


Fig. 4 – Groups and Keys

A group is a pool of one or many users. It is featured by a group communication key, which will be used to generate session communication keys for that group, and a set of security policies (e.g., cryptographic algorithm used to protect the information related to that group and mechanism to generate the session keys) which will be used to decide if the users are entitled to belong to that group.

As shown in Fig. 4, Group A is made up of three users (User 1, User 2, and User 4) whilst Group B is made up of four users (User 1, User 2, User 3 and User 5). The groups are usually created by a security administrator (managed groups). Nevertheless, groups can also be created manually by the users (manual groups), according to the specific organization security policies. In both cases the security architecture is the same.

Whenever a group is created, a group communication key is automatically generated by the SEcube™ and every user receives all the communication keys related to the groups which it belongs to. The SEcube™ platform provides APIs for the secure distribution of group keys to the entitled users.

Referring to Fig. 4, it is easy to understand that two or more users can access the information only if they share at least one group communication key. For example, User 1 and User 3 can access the same information, since they belong the same group (Group B) and, accordingly, they share the same key (Key B). When users share more than one key (they have more than one group in common) the key related to the smallest group is selected, since it is known by less users (more secure). As detailed in paragraph 6, this mechanism is part of the negotiation process, which is executed every time a secure communication link is established.

On the programming side, every group is identified by a unique number (GroupID). Each SEcube™ can manage up to 256 communication groups. In addition, there are two special groups: *personal* and *family*. The *personal group* is associated to the user, which owns the SEcube™ based device. It is used to manage personal, non-shareable information and its GroupID is made up of all zeroes. The *family group* is associated with all of the users inside an organization. It is used to manage information which can be shared with everybody and its GroupID is made up of all ones.

This approach allows developers and users to focus on the final secure service, since they are not required to be familiar with keys and algorithms (e.g., kind of key, key size and algorithm type). The cryptographic complexity is easily and transparently managed by the SEcube™ platform according to the security policies set at provisioning time by a security administrator. On the other side, cryptographic experts and developers are free to customize any part of the system, thanks to the open source nature of the platform.

5 SE*file*, the SEcube™ based Secure File System

Data at rest can be easily protected through the SE*file* technology, a Secure File System level-2 library which allows standard applications to access standard file systems through the SEcube™ cryptographic layer, performing file encryption, signature, and name remapping.

There are several technologies to protect file systems. For example, the ORI File System [6] aims to provide a security solution for distribute file systems, replicating mechanisms like multi-user cloud like drop box and version control like Git. Another example is the Secure File System module [7], which is implemented at Linux kernel level to encrypt any folder with encrypt prefix. All these solutions depend on specific implementations (e.g. operating systems, file systems, etc.) and sometimes are more focused on extra functionalities (e.g. cloud, versioning, etc.) than security. For these reasons they may be very invasive, forcing the users to change their habits, and at the same time they are not so portable. On the contrary the SE*file* technology is independent of the other layers (e.g. operating systems, file systems, etc.). It can be easily integrated in any pre-existing systems (e.g. drop box) and allows developers to create very low-impact and portable security solutions focusing on the content protection.

As shown in Fig. 5, SE*file* operates as a transparent virtual layer on top of the regular file system, providing a set of multi-paradigm, Level-2 APIs (SFS_Open, SFS_Read, SFS_Write, SFS_Seek and SFS_Close). Since the SE*file* is implemented on the top of the standard file systems functions, it is independent of the Operating System.

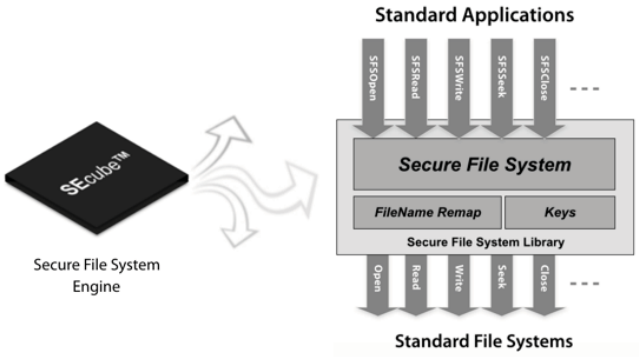


Fig. 5 – SE*file* library concept

Being open source, the SEcube™ implementation can be verified or improved at any time. Nevertheless, the developers that do not feel comfortable with cryptography, or simply trust the open source community system validation, can benefit from the security abstraction level, which simplifies all the cryptographic mechanisms.

For example, when a secure file is created through the SFS_Open function, it is just required to specify the protection scope: for me only (personal), for a group of two or more (group), for everybody (family). When the protection scope is a group, a group identifier must be provided as described in section 4. By doing so, users and developers do not need to deal with keys, algorithms and other low level complex cryptographic features, which are transparently managed inside the SEcube™.

As shown in Fig. 6, a secure file is made up of many encrypted and signed sectors. The first sector is partially encrypted, since it contains the secure file header, which includes some clear and coded fields (e.g. initial vector and security scope). All other sectors are fully encrypted. Each sector is 512 Bytes long in order to be read/written atomically and prevent possible data corruption issues, especially on mobile devices (e.g., low battery).

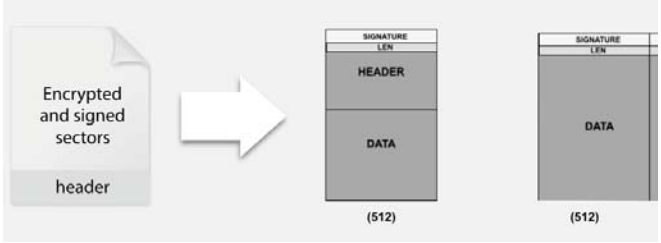


Fig. 6 – Secure File structure

As shown in Fig. 7, when a secure file is created, the file name is coded in a way that nobody can recognize it looking directly at the physical file system. For example, the coded file name can be calculated inside the SEcube™ by a one-way digest function (e.g., SHA256) of the real file name in combination with an SEcube™ common information (family secret), whilst the real file name is encrypted in the secure file header. In this case a folder containing secure files looks like:

…/B5B8D0F121F9596A…4BBC4829615B968EDA.se
…/B5A6135EF9B1783C…F981BC7AA9F541D93F.se
…/B5E93F7BC1DD18D…8AF3C6BA6135EF9B17.se

According to the library paradigm, SE*file* can be provided with different implementations, such as hardware cryptography (accelerated implementation on the embedded FPGA), software cryptography (firmware implementation on the embedded CPU), or extra features (multi-level secure cache, anti reply attack, auto rescue, etc.).

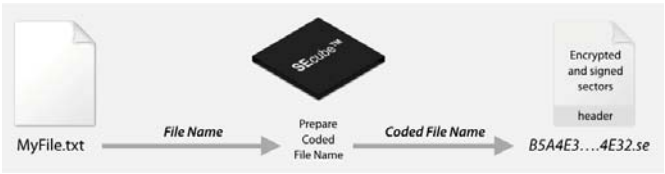


Fig. 7 – Coded file name

Beside some basic posix style APIs, the SE*file* library provides file management functions like SFS_GetFileList (return the list of secure files in a specific folder showing their real file names) and SFS_DeleteFile (delete a specific secure file).

The simple set of posix-like APIs makes SElink easy to be integrated in third-party libraries or applications. As shown in Fig. 8, an interesting library integration example is the combination of SE*file* and SQLite.

SQLite offers a virtual file system interface to interact with the underlying operating systems, thus integrating SE*file* it can be provided with a strong, fast and customizable security layer to the host environment. This approach is very efficient: it keeps the SQLite interface unaltered and no changes are required at the application level.

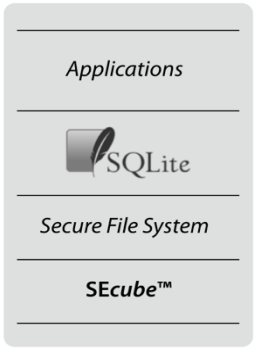


Fig. 8 – SE*file* and SQLite integration

SE*file* can be also combined with third-party solutions to generate secure application and services. For example, SE*file* can be used in a Dropbox™ folder to deploy a zero-impact secure cloud solution. In a similar way, SE*file* may be used in combination with an email client to store mails and attachments. Although the same technology may be used to send and receive files, the next section presents a more effective and flexible library to protect data in motion.

6 SE*link*, the SEcube™ based Secure Link

Whenever an information item is transferred among entities, the connection links offer an attractive opportunity for attackers to catch sensitive contents.

There are many solutions which provide security for the most standard communication channels, like TLS [8] for IP based data transfers, Virtual Private Networks (VPNs), etc. Nevertheless, all these solutions are implemented for specific communication channels, sometimes they are complex, their overhead is not acceptable and usually provide point-to-point only security.

In line with our general approach, we propose a solution which is very light, easy to be integrated in any environment (over any protocol), open source, portable, multi-device and multi-paradigm: SE*link*.

The SE*link* technology allows protecting data in motion on both point-to-point and point-to-multipoint links through the SEcube™ cryptographic layers, which performs negotiation, encryption, and signature operations. It is a security layer running on top of any transport technology (e.g. IP, TCP, HTTP/S, SMTP, XMMP, CoAP and custom protocols) and featured with a very low overhead in order to be fast and easily integrated.

When two or more entities need to set up a secure communication channel to exchange data, the SE*link* library negotiates the session keys and performs encryption, signature, decryption and verification operations on the exchanged data.

As shown in Fig. 9, in the first phase the SE*link* performs a negotiation process checking the security policies (e.g., methods and master keys to derive session communication keys), agrees the common parameters (e.g., cryptographic algorithms, predefined communication groups/ keys, etc.) and derives the final session communication keys. After performing a successful negotiation, the secure link is ready for data transfer and the negotiated keys can be used to encrypt, sign, verify, and decrypt the information.

Fig. 9 – SE*link* point-to-point example

Negotiation can be duplex or simplex. The duplex negotiation must be executed between two entities (peer-to-peer) that will be co-responsible in generating the final session communication keys. Other entities can join the link at a later stage (peer-to-multi-peer) performing a simplex negotiation, which requires a master entity, already in the link, and one or more slave entities willing to join the link. The master entity pushes negotiation parameters that allows slave entities to generate the communication session keys already negotiated. The generation is possible only if the slaves are entitled to join the link (e.g., security policies match and same communication group).

The negotiation process is performed in two messages, only. For example, in the case of HTTP/S links, the negotiation is performed in one standard GET method.

SE*link* provides a small and easy way to use set of multi-paradigm, Level-2 APIs to manages both binary data, suitable for protocols like IP, and text based data, suitable for protocols like HTTP. In order to simplify the integration on stateless systems (e.g., web servers) SE*link* functions are multi-sessions and the internal state can be saved outside the SEcube™ platform in a secure (it is encrypted and signed) and private (it can be reused by the originator SEcube™ only) container automatically generated by the library. As shown in Fig. 10, each entity is able to concurrently manage several SElink channels. In this case, A and C work on two secure links at the same time and three total sessions are negotiated: A-D, A-C, and B-C.

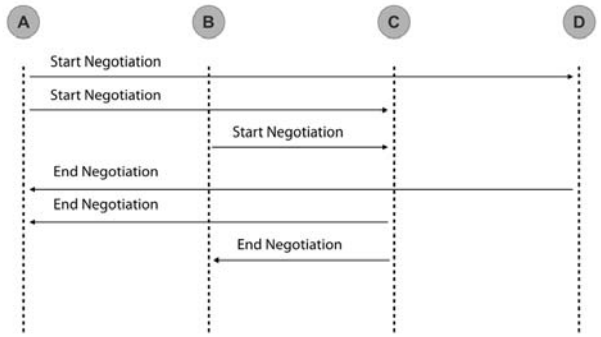


Fig. 10 – SE*link* multi-session negotiation

Similar to the SE*file* technology, SE*link* can be easily combined with third-party solutions to generate secure application and services. For example, SE*link* can be used in combination with XMPP based messaging applications in order to protect text and attachments.

The SE*link* stateless nature makes it suitable for operations on two or many channels even when they are related to the same service. For example, in case of VoIP services, SE*link* can start the negotiation procedure on the signaling channel. After successfully completing the process, the communication stream (e.g., voice) can be encrypted, signed, decrypted, and verified on the data channel, which is logically (and sometime physically) different from the signaling one.

7 Conclusions

This paper introduced some of the SEcube™ platform based methodologies to protect both data at rest and data in motion, which covers any security services and solutions.

It is possible to combine flexibility in the choices of technology and in the level of security to be attained with a well-organized and modular protection of data at rest and data in motion. As described in [2] and detailed in [3] for the software development platform, in [4] for the hardware and cryptography, and in [5] for the knowledge management via properties, also in security, the overall platform is developed to be robust and versatile, espousing the principle of models for encapsulation and model assembly for application development.

This approach delivers both the desired cost containment and the change friendliness.

Technologies like SE*file* and SE*link* provide a flexible security layer which can be deployed through several devices (multi-device) according to the target environment. The SEcube™ platform can be easily integrated in pre-existing systems thanks to a simple set of APIs that provide several entry points (multi-level). The internal implementation can be tuned (multi-paradigm) in order to match requirements of security, costs and user experience choosing among several options: from a standard and software only implementation to a full-custom hardware solution.

In the previous sections we described how to manage several use cases by means of the SE*file* and SE*link* technologies individually. Nevertheless, their combination is even more effective, since it is possible to address complex scenarios like remote machines and web based services, providing a large protection layer against server, client and communication side attacks.

All the above features make the SEcube™, and its associated abstraction layer and libraries, a unique security platform, which provides a multi-level, multi-device, multi-paradigm solution to realize high-grade security services and applications minimizing the development effort and reducing drastically the time-to-market to secure products.

8 Acknowledgment

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie)

9 References

[1] A. Varriale, E. I. Vatajelu, G. Di Natale, P. Prinetto, P. Trotta, and Tiziana Margaria, "SEcube: An Open-Source Security Platform in a Single SoC," Proc. 11th IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS 2016), April 2016, Istanbul, Turkey

[2] A. Varriale, G. Di Natale, P. Prinetto, B. Steffen, and T. Margaria, "SEcubeTM: An open security platform: General approach and strategies," Proceedings of the International

Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[3] S. Boßelmann, J. Neubauer, S. Naujokat, and B. Steffen, "Model driven design of secure high assurance systems: an introduction to the open platform from the user perspective," Proceedings of the International Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[4] G. Di Natale, A. Carelli, P. Trotta, and T. Margaria, "Model driven design of crypto primitives and processes," Proceedings of the International Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[5] G. Airò Farulla, M. Indaco, A. Legay, and T. Margaria, "Model driven design of secure properties for vision-based applications: A case study," Proceedings of the International Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[6] Ali Josè Mashtizadeh, Andrea Bittau, Yifeng Frank Huang, David Mazières, Stanford University, Ori File System Web Site, http://ori.scs.stanford.edu/

[7] Rajesh Kumar Pal, Indian Institute of Technology, Secure File System Thesis

[8] T. Dierks, E. Rescorla, Network Working Group, The Transport Layer Security (TLS) Protocol Version 1.2, https://tools.ietf.org/html/rfc5246

MODEL DRIVEN DESIGN OF SECURE HIGH ASSURANCE SYSTEMS: AN INTRODUCTION TO THE OPEN PLATFORM FROM THE USER PERSPECTIVE

Steve Boßelmann and Johannes Neubauer and Stefan Naujokat and Bernhard Steffen

Chair of Programming Systems
TU Dortmund
{name.surname}@cs.tu-dortmund.de

ABSTRACT

We present DIME, an integrated solution for the rigorous model-driven development of sophisticated web applications based on the Dynamic Web Application (DyWA) Framework, that is designed to flexibly integrate features such as high assurance and security. DIME provides a family of Graphical Domain-Specific Languages (GDSL), each of which tailored towards a specific aspect of typical web applications, including persistent entities (i.e., a data model), data retrieval (i.e., search queries), business logic in form of various types of process models, the structure of the user interface, and security. They are modeled on a high level of abstraction in a simplicity-driven fashion that focuses on describing *what* application is sought, instead of *how* the application is realized. The choice of platform, programming language, and frameworks is moved to the corresponding (full) code generator which may be changed without touching the models leading to high assurance systems.

1. INTRODUCTION

The DIME approach is a consequent refinement of the realization of jABC4 [1] for process modeling and DyWA [2] for data modeling empowering prototype-driven application development. In the spirit of its predecessors DIME follows OTA (One Thing Approach) [3] and XMDD (Extreme Model-Driven Design) [4] and puts the application expert (a potential non-programmer) in the center of the development process. Hence, the different aspects of an application are described with the most adequate form of model, respectively. All these models are interdependently connected shaping the one thing in a very formal yet easy to understand and use manner to the extend that it can be one-click-generated to a running product. DIME can be used to realize a wide range of web applications. We are just starting to explore its potential. Central design goals on this journey are simplicity [5] and agility [6] as well as security and quality assurance.

DIME enables user-level development of sophisticated web applications. The user starts with the designs of various graphical models that cover different aspects of the target application. These models form the input for a subsequent product generation step in which the full target application is assembled from a variety of generated files that contain the respective source code. The target of this product generation is the DyWA framework that fosters the prototype-driven web-application development throughout the whole application life-cycle in a truly service-oriented manner [7]. In short, modeling and code generation is done in DIME whereas DyWA supports the product deployment phase, constitutes the actual runtime environment and manages data persistence. Furthermore DyWA explicitly enables and supports continuous evolution, the sense of continuous model-driven engineering [8], in a rigorous manner, which facilitates ensuing iterations through the product re-design, re-generation and re-deployment cycle.

The DIME-approach provides the user with both an early prototype of an up-and-running web application from the very beginning of the development process as well as explicit support for product evolution due to the agile nature of version management regarding data handling and persistency by the DyWA framework. Altogether, the approach has the potential to tremendously push development cycles in an agile but consistent manner which even comprises security aspects.

2. MODELING ENVIRONMENT

For the model design phase, DIME provides a family of Graphical Domain-Specific Languages (GDSL), each of which tailored towards a specific aspect of typical web applications. These span *Data models* for the design of domain models, *GUI models* to specify the structure of (re-usable components of) web pages as well as different types of *Process models*, each of which tailored towards specific aspects of a web application's behavior. The apparent relations between each of these aspects is modeled by means of cross-referencing to create hierarchical model structures. For this purpose, DIME provides SIBs (Service Independent Building Blocks) in terms of basic model components that link to existing models or to atomic components. SIBs are essential for the effective realization of the omnipresent concept of model

ISBN: 1-60132-445-6, CSREA Press ©

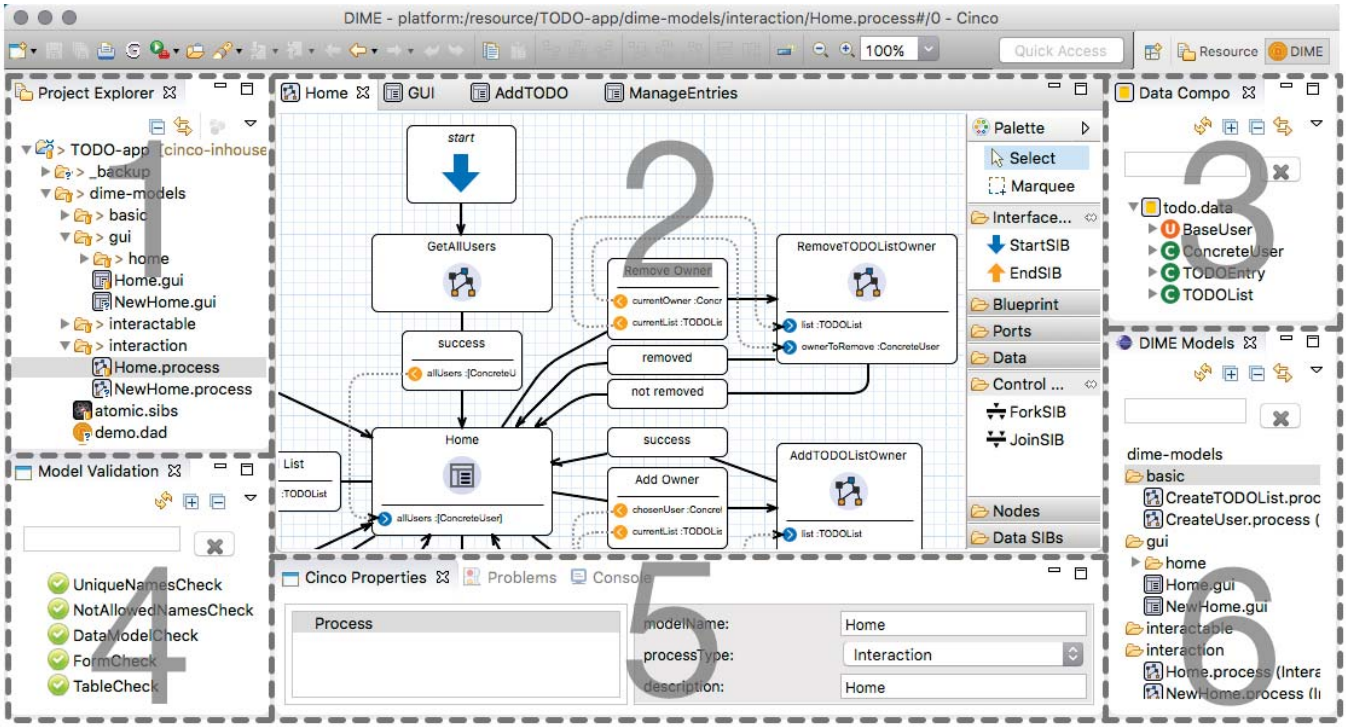


Fig. 1. User interface of DIME with exemplary arrangement of views: (1) Project Explorer. (2) Model Validation View. (3) Diagram Editor with Palette. (4) Properties View. (5) Data View. (6) Models View.

re-use.

The user interface of DIME has been specifically tailored towards supporting the recurrent modeling steps, i.e. to provide guidance for the user by means of quick access to available model entities and relevant properties. The following sections provide a short overview of the DIME user interface as well as more details about available modeling languages and the structural properties of the various models.

2.1. The DIME Application

DIME is a desktop application based on the Eclipse Rich Client Platform (RCP), developed with the CINCO SCCE Meta Tooling Framework [9] that facilitates the development of domain-specific graphical modeling tools in a rigorous model-driven fashion. As RCP application, it consists of a set of plugins for the Eclipse framework that provide support for effective model editing and specific views on the current workspace. Models in DIME are foremost graph models formed from nodes and (directed or undirected) edges between them. The provided views take on this inherent structure and provide dense overviews, quick access or other kind of design-relevant information for the user. Fig.1 exemplarily depicts an arrangement of these views constituting the user interface of DIME. Apart from the generic Project Explorer listing files in the workspace, each of the DIME-specific views is shortly introduced in the following.

The **Diagram Editor** in the center of the DIME interface provides the canvas to draw the various graphical models on. Additionally, it provides a palette with basic model components that are specific for the type of the model that is currently opened and shown in the editor.

The **Models View** provides the user with a dense overview of available Process models in the current workspace. From this view, a model can be dragged and dropped on the canvas in the diagram editor iff the currently edited model supports cross-references to models of that respective type. This action triggers the creation of a new node inside the currently edited model that holds a reference to the existing model in the workspace. This is the essence of model re-use in the context of DIME.

The **Data View** lists data models in the current workspace. In particular, data types and type attributes are enlisted in a tree-based structure with the respective data model as root element. Data types can be dragged and dropped into special data containers of other models, in order to introduce variables for data exchange between the model components.

The **Properties View** provides access to attributes and parameters of nodes and edges inside the currently edited model. This is where parameter values for these entities can be changed by the user. Available attributes and parameters differ depending on the type of the respective model component.

The **Model Validation View** lists the results of syntactic and semantic checks that are applied to the currently edited model. These checks are specifically tailored towards the type of the respective model and dynamically evaluated at model design time. It provides guidance for the user and facilitates correctness of the model, by listing warnings and errors with respect to the affected entity or model substructure. Model validation in DIME spans various aspects comprising the enforcement of unique names, the correct use of expression languages, identification of missing edges, various syntactical requirements.

2.2. Graphical Modeling Languages

Each model type in DIME is a well-defined graphical modeling language that relies on nodes and edges as the basic components of graph models, as well as containers that are special nodes that again can contain other nodes. Graphical modeling is done by means of drag-and-drop operations in interaction with the canvas of the Diagram Editor. Basic model components are dragged from the editor's palette and dropped on the canvas. This triggers the creation of a new node in the current graph model representing an instance of the respective component.

Existing nodes may be connected via edges in a drag-and-drop manner, i.e. clicking on the source node and dragging an intermediate line to the target node. If multiple edge types are suitable for connecting the respective nodes, a selection dialog is shown for the user to select the desired type.

Due to limited space of this article, the syntax and the semantics of the DIME's various model types cannot be described in every detail. In the following the most important concepts in this context are introduced.

Service Independent Building Blocks (SIB) are basic model components in DIME that are essential for the effective realization of the omnipresent concept of model re-use by means of cross-referencing and - as a special case of that - the creation of hierarchical model structures [10]. In essence, SIBs either provide a link to an existing model or to an atomic model component. Atomic in this context means that the component is self-contained and integrated in a service-oriented fashion and not based on models within the current workspace.

The modeling operation to create SIBs by means of nodes in the currently edited graph model is done by means of drag-and-drop operations in interaction with the canvas of the Diagram Editor. It is basically the same operation as described in the context of basic model components, but the respective model is dragged from the Data View or Models View, depending on its actual type. Dropping it on the canvas triggers the creation of a SIB by means of a node that holds a cross-reference to the respective model.

In the following the various model types are discussed briefly.

Data models in DIME allow for the graphical design of domain models based on common data modeling concepts in terms of classes and attributes as well as relations (uni- or bi-directional associations) between them. The structure of Data models reflects the data structures that are manageable by DyWA, as the latter maintains data objects and provides support for persistence at runtime.

GUI models are used to specify the structure of (re-usable components of) web pages that make up the user interface of the target web application. Hence the structure of GUI models reflects the structure of web pages in order to enable user interface design in a familiar manner.

Process models in DIME allow for the graphical definition of the business logic of the target application. In particular, in DIME exist different types of process models. Each of them is tailored towards specific aspects of a web application's behavior. At design time, it depends on the actual types of the involved models whether cross-references are allowed and how they are handled. On the other hand, the syntax of the different types of process models is nearly the same. The common syntactical features are going to be explained in the course of the discussion of process modeling in Sec. 3.

DAD model. The DyWA Application Description (DAD) model is used to specify the entities that are relevant for the application runtime. A suitable configuration comprises the declaration of relevant domain models, an interaction process that provides the landing page for the target web application and an optional startup process to be invoked when the application is started. This configuration is the entry point for the product generation phase in which source code of the target web application is generated.

3. PROCESS MODELING

Process models in DIME comprise both, a control flow aspect as well as a data flow aspect. In the following the main design concepts regarding each of these aspects are described. In this context, we provide figures that contain models of an ongoing exemplary *TODO-app* application that basically manages lists of TODO entries for its users. The interested reader may find further informations as well as a detailed tutorial on the DIME website[1].

3.1. Control Flow

Process models contain a single start node and might have multiple end nodes. In between, the control flow is modeled by means of connecting multiple SIBs via directed control flow edges. In this context, SIBs can not only hold references to other Process models (i.e., Process SIBs) but also to GUI models (i.e., GUI SIBs). While integrating Process SIBs fosters model re-use by means of sub-processes, integrating GUI

[1] http://dime.scce.info

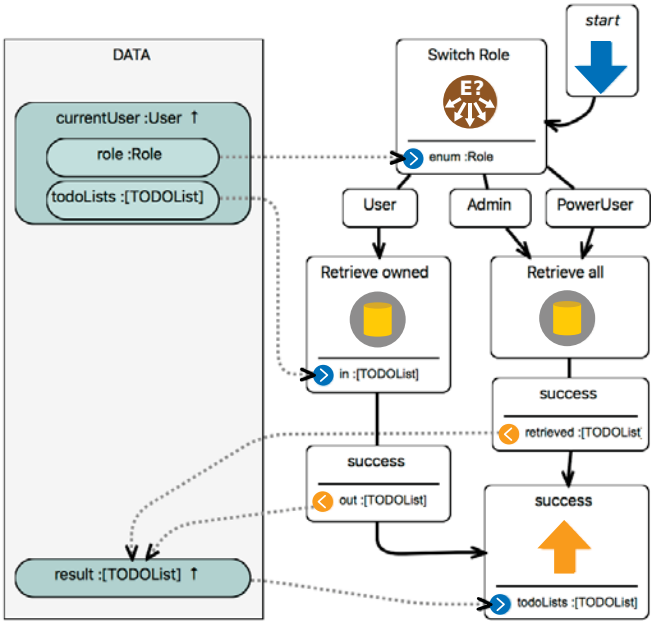


Fig. 2. Process model *GetTODOLists*

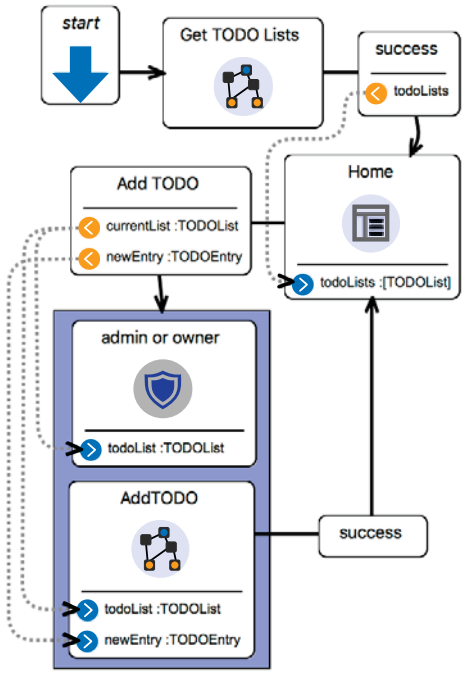


Fig. 3. Process model *AppHome*

SIBs into a process expresses that at runtime throughout the execution of this process an interaction with the user of the application has to take place. In both cases, the subsequent control flow might depend on the actual outcome, be it an execution result of the sub-process or eventual input provided by the user, respectively. In order to reflect this at model design time, the concept of *Branches* is introduced.

SIBs as components of Process models consist of a node and multiple so-called Branches. While the node represents the actual activity to be executed, each Branch represents one possible outcome of this execution. In particular, the control flow follows only one Branch of a SIB at a time. As an example, Fig. 2 shows a Process model that contains a SIB labeled *Switch Role* with three Branches represented by outgoing edges labeled *User*, *Admin* and *PowerUser*. While in this example the SIB represents an activity that identifies the actual role of the current user of the application, its Branches cover all possible cases. It is also apparent from Fig. 2 that the subsequent control flow depends on the actual Branch taken at runtime. For GUI models each user interaction with the respective web page is interpreted as a branch, e.g., clicking a submit-button in a form or following a hyperlink to another page. As an example, the GUI Model depicted at the top of Fig. 5 is integrated into the Process model *AppHome* in Fig. 3 by means of a GUI SIB represented by the node labeled *Home*. Consequently, the button of the form depicted in Fig. 5 bottom is mapped on the single Branch *Add TODO* of the GUI SIB.

As different outcomes of a SIB might convey different provided data, there is also a data flow aspect in the context of Branches to be discussed.

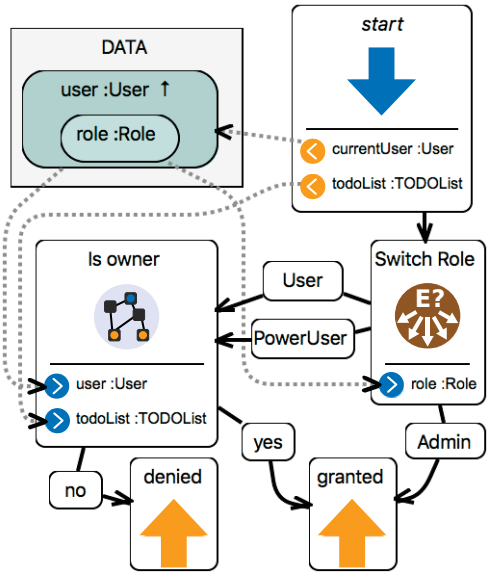


Fig. 4. Security Process model *AdminOrOwner*

3.2. Data Flow

Data flow in the context of Process models too is modeled in a graphical manner. For this purpose, the concepts of ports as well as the data context are introduced.

For modeling access to the runtime context of the application, Process models provide a specific *DATA* container that holds the respective data objects, i.e. Data SIBs with references to data types specified in respective Data models. In the following, this container is referred to as data context of the Process model. In order to enable modeling of the actual data flow, SIBs in the context of Process models can have so-called *Input Ports* while Branches in turn can have *Output Ports*, representing data input and data output, respectively. In Fig 2, each of the *success*-Branches of the retrieval activities *Retrieve owned* and *Retrieve all* have one Output Port representing the retrieved TODO lists. Connecting any of these ports with the Data SIB *result* in the data context via a data flow edge expresses that at runtime this context object is provided by the Output Port of the respective Branch. As both the data output of the Branches as well as the data object in the data context are typed as list of `TODOList` objects, the data flow edge is valid. In turn, invalid data flow edges are recognized and prevented by the DIME framework. Additionally, Fig. 2 shows how the value of the data object labeled *result* in the data context is provided as data for the Input Port *todoLists* of the end node *success* of the Process model via a data flow edge. Altogether, the process depicted in Fig. 2 describes the activity of retrieving the TODO lists owned by the current application user, or all TODO lists from a database in case that this user has a role with special privileges.

If Process models are integrated into another model by means of a Process SIB, each of its end nodes is mapped on a separate Branch of this Process SIB. Furthermore, each Branch of the Process SIB would have an Output Port for each Input Port of the corresponding end node. As an example, Fig. 3 shows the Process model *GetTODOLists* as shown in Fig. 2 integrated into the Process model *AppHome* by means of a Process SIB labeled *Get TODO Lists* in the figure. Note that this Process SIB has a Branch *success* according to the end node of the Process model in Fig. 2 and this Branch has an Output Port *todoLists* related to the Input Port of the respective end node with the same name.

Though not depicted in this figure, the same concept applies for Output Ports of start nodes in relation to Input Ports of corresponding Process SIBs. These are referred to as model parameters. The overall approach is directly related to the concept of formal and actual parameters of functions in programming languages.

In the context of GUI SIBs, Output Ports of Branches are related to data objects in the respective GUI model. As an example, the data objects connected via edges labeled *Submit* to the button in Fig. 5 bottom match the Output Ports of the corresponding Branch *Add TODO* of the GUI SIB in Fig. 3

3.3. Security Guards

Security guards is a concept in DIME to handle access to data objects based on a special type of Process models named *Security Process*. The task of a Security Process is to decide upon whether the current user of the target application fulfils specific criteria in relation to the respective input of the process. Hence, models of this type follow a predefined structure that requires two end nodes *granted* and *denied*, as well as a model parameter named *currentUser* with respective user type. Fig. 4 shows an example of a Security Process that realizes the decision upon whether the current user is admin or owner of a TODO list specified as model parameter. The depicted process in particular follows the structural requirements discussed above.

This Security Process can be used to decide whether the current user is allowed to manipulate a specifc TODO list. Fig. 3 depicts its integration into the *AppHome* process. It is used to restrict the manipulation of a TODO list by means of adding new TODO entries. The Process SIB *AddTODO* is contained in a so-called *Guard Container* together with the Security Process *AdminOrOwner* (in this context a *guard process*) to express that the execution of the first needs to be guarded by the latter. The underlying security concept is discussed in the following section.

4. SECURE HIGH ASSURANCE SYSTEMS

There are two different views from which we would like to consider the security and assurance aspects of applications built with DIME: model-level, and platform-level.

4.1. Model-Level

We follow a thorough modeling approach with DIME. Its interdependent models of various types are each tailored to the different aspects of a web application. Altogether they build one contiguous specification with all the necessary information to generate a fully operational web application in a service-oriented way on any platform choosing arbitrarily from the set of adequate frameworks. Since we obtain a coherent description of our application, the barriers between the layers of an implementation of such an application diminish. Like in aspect-oriented programming we are able to describe a property of our application only once and the analysis during code-generation ensures, that this property holds for all layers, beginning with the Javascript code in the browser, reaching to the business-logic and persistence layer in an application server.

This way, error-prone replication of check-code to all the layers becomes unnecessary. Therefore, we gain high assurance that the system behaves as expected. In addition, this approach can be permeated to dynamic access-control rules leading to secure systems. Of course, the generators may have flaws, but on the one hand the generators can be reused

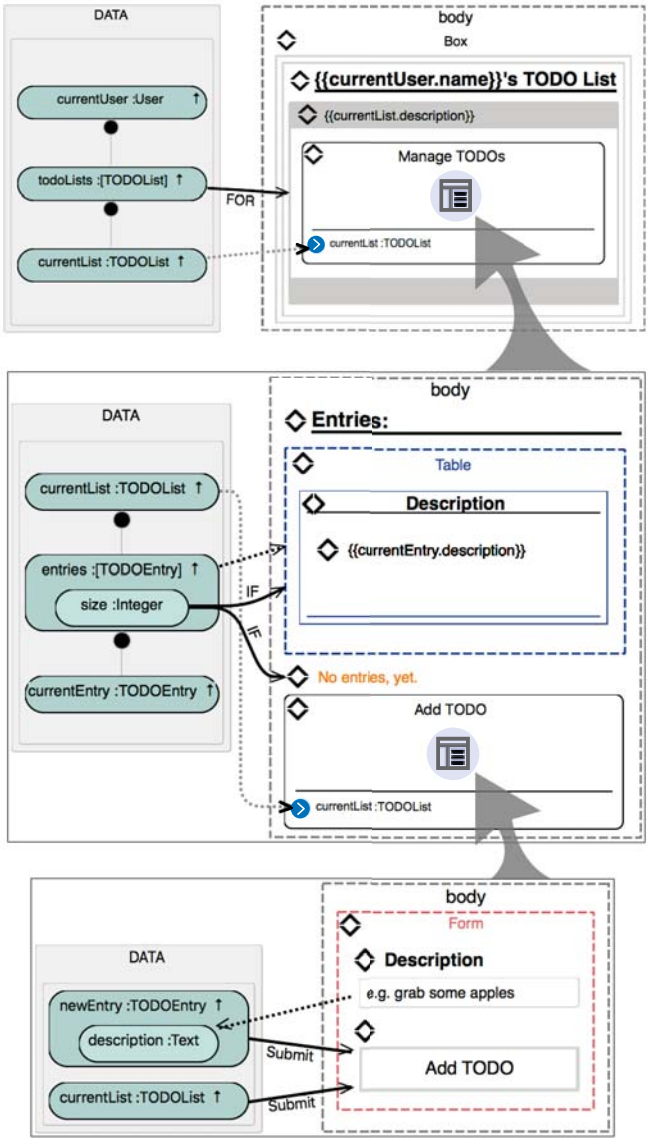


Fig. 5. Hierarchical GUI models of the TODO-app

for many different applications, and therefore it is much more likely that these flaws are revealed than in a single manual implementation. On the other hand, fixing such an issue can be carried out to all applications built with DIME, just by generating the applications again.

Referring to the TODO-App example application in Fig. 3 the process *AddTODO* is secured with the guard process *admin or owner*. This is the only place where this property has to be set. The analysis during code generation can follow the control flow back to the branch *Add TODO* of the GUI-SIB *Home*, and then traverse the cross-references (see Fig. 5 **top** to **bottom**) to the corresponding GUI model *Add TODO GUI* (cf. Fig. 5 **bottom**). It may then generate a case differentiation for the user interface, which will disable the elements of the corresponding form (see red rectangle with label "Form"

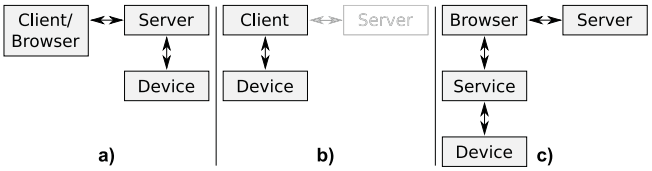


Fig. 6. Variants of server-side encryption via a dedicated hardware device.

in Fig. 5 **bottom**) or omit it completely, if the guard process evaluates to *denied* for the respective TODO-list. Since the TODO-Lists are rendered in a `FOR`-loop (cf. Fig. 5 **top**), a backend service can be generated, that evaluates the guard process *admin or owner* for the TODO-lists to be shown all at once and returns which may be edited and which not. Further on, the guard process will be called each time before the process *Add TODO* is executed and the TODO is added only, if the guard evaluates to *granted*. This means, even if someone calls the process directly it is assured, that the access control rules are satisfied.

4.2. Platform-Level

Classical client-server architectures often use server-side encryption of content. This may be supported by a hardware device as shown in Fig. 6 **a)** or done completely in software. Supporting such a solution with high assurance entails the need for some expertise in cryptography in the development team and for each new application there is the potential to make mistakes. Using a generative approach can lower this risk and reduce the responsibility of a development team.

Since the DIME approach is service-oriented, not everything is generated down to the last statement. Instead, a base system (i.e., DyWA) is used, which is a manually implemented full-fledged web application right from the start. Only the data-structures, business-logic (including access control), and the structure of the user interface are generated on top of modern technologies like *Angular 2*[2] and *Java EE*[3]. The modeling-level is completely independent from the environment it is generated to. Hence, it is very easy to integrate support for cryptography on the server-side and reuse this for every DIME application.

Furthermore, in the last years the interest to *protect content against the provider of a service* has grown tremendously. For sure, several uncoverings of data privacy violations, by employees of providers, hackers, and even governments contributed to this trend. Several cloud services and instant messaging services therefore introduce *end-to-end encryption*, so that the provider will not be able to access the data – as it

[2]https://angular.io
[3]https://www.oracle.com/java/technologies/java-ee.html

would be the case with server-side encryption – neither willingly or under pressure, even, if it is stored on their servers.

End-to-end encryption has a weak spot: the client. This can be both the user in front of a device who in general is not an expert in cryptography and the device itself. There are several approaches to increase trustability of devices on the software-level [11, 12] as well as on the hardware-level [13], leading to architectures as shown in Fig. 6 **b)**.

Securing web-applications with a desktop client via end-to-end encryption is already challenging, but using hardware-supported encryption in a web-application in combination with a browser client is even harder. This is ironically due to limitations of Javascript that have been introduced for security reasons in the first place. The Javascript interpreter lives in a so called sandbox and is not allowed to (freely) access devices on the local machine like the file system or the hardware security device.

Since we describe only *what* an application is sought to do, we can transparently add a web service to our setup (see Fig. 6 **c)**) running on the local machine, which is able to encrypt and decrypt arbitrary content using a hardware device like the SEcube [14] and thereby guaranteeing end-to-end encryption of Web applications. The code-generator then adds (Javascript) code to the browser client calling this web service to encrypt the content (e.g., the title and description of a TODO entry) before it is sent to the server and to decrypt it before it is presented in the web page to the user. This way, the sandbox limitations are circumvented in a clean and simple way.

The server stores the encrypted content in the database and returns it on demand. There is no need that the server is aware of the encryption. In more complex scenarios with multiple receivers, e.g. a shared TODO-list, a protocol for the exchange of the respective public keys needs to be generated into the application in advance.

The setup of the service and the hardware device, can be integrated into the authentication process of the web application. The service itself can be shipped as an executable on a mass storage device integrated into the security hardware, digitally signed with the private key in the hardware itself. The complete communication between the Javascript client in the browser and both the local encryption service and the server-side of the web application will be secured via HTTPS to prevent man-in-the-middle attacks for getting one's hand on the meta-data which cannot be end-to-end encrypted, since the server needs to be able to process it. This way, using a DIME application can be secure even on a public computer (e.g., in an internet cafe).

5. SUMMARY AND OUTLOOK

We have presented DIME, an integrated solution for the rigorous model-driven development of sophisticated web applications that is designed to flexibly integrate features such as high assurance and security via a family of Graphical Domain-Specific Languages (GDSL), each of which tailored towards a specific aspect of typical web applications, including persistent entities, data retrieval, business logic, the structure of the user interface, dynamic access control, and security. DIMES simplicity-driven modeling approach makes the choice of platform, programming language, and (security) frameworks transparent, by moving them to the underlying (full) code generator which may be changed without touching the models.

6. LITERATURE

[1] Bernhard Steffen et al. "Model-Driven Development with the jABC". In: *HVC 2006, Haifa, Israel*. Vol. 4383. LNCS. Springer, 2007, pp. 92–108.

[2] Johannes Neubauer et al. "Prototype-Driven Development of Web Applications with DyWA". In: *Proc. of 6th ISoLA*. LNCS 8802. Springer, 2014, pp. 56–72.

[3] Tiziana Margaria and Bernhard Steffen. "Business Process Modelling in the jABC: The One-Thing-Approach". In: *Handbook of Research on Business Process Modeling*. IGI Global, 2009.

[4] Tiziana Margaria and Bernhard Steffen. "Service-Orientation: Conquering Complexity with XMDD". In: *Conquering Complexity*. Springer, 2012, pp. 217–236.

[5] Maik Merten and Bernhard Steffen. "Simplicity driven application development". In: *Journal of Integrated Design and Process Science (SDPS)* 16 (2013).

[6] Tiziana Margaria and Bernhard Steffen. "Simplicity as a Driver for Agile Innovation". In: *IEEE Computer* 43.6 (2010), pp. 90–92.

[7] Tiziana Margaria, Bernhard Steffen, and Manfred Reitenspieß. "Service-Oriented Design: The Roots". In: *Proc. of 3rd ICSOC*. Vol. 3826. LNCS. Springer, 2005, pp. 450–464.

[8] Tiziana Margaria and Bernhard Steffen. "Continuous Model-Driven Engineering". In: *IEEE Computer* 42.10 (2009), pp. 106–109.

[9] Stefan Naujokat et al. "CINCO: A Simplicity-Driven Approach to Full Generation of Domain-Specific Graphical Modeling Tools". To appear in STTT (2016).

[10] Bernhard Steffen et al. "Hierarchical Service Definition". In: *Annual Review of COMMUN ACM* 51 (1997), pp. 847–856.

[11] Giorgio Di Natale et al. "Model driven design of crypto primitives and processes". In: *This volume*. 2016.

[12] Roberto Baldoni and Luca Montanari. "Italian National Cyber Security Framework". In: *This volume*. 2016.

[13] Antonio Varriale et al. "SEcube^{TM}: Data at Rest & Data in Motion protection". In: *This volume*. 2016.

[14] Antonio Varriale et al. "SEcube^{TM}: An open security platform: General Approach and Strategies". In: *This volume*. 2016.

Towards Model Driven Design of Crypto Primitives and Processes

Alberto CARELLI*, Giorgio DI NATALE†, Pascal TROTTA‡, Tiziana MARGARIA‡
*CINI Cyber Security National Lab, Rome, Italy - alberto.carelli@Consorzio-CINI.it
†LIRMM, CNRS, Montpellier, France - giorgio.dinatale@lirmm.fr
‡University of Limerick Lero - The Irish Software Research Centre, Limerick, Ireland - tiziana.margaria@lero.ie

***Abstract*—To be understandable and reusable at large scale, also by non-experts in security, Crypto primitives must be implemented in a modular way, and come with well organized and well described processes to help understanding, foster adoption, and ensure a proper embedding in the applications they must protect. In this paper, we reap the benefits of the modular hardware and software architecture of the SEcube, and lift the issue of crypto-primitives management from the traditional code level to a model driven approach. On small examples, we illustrate the essential features of the approach concerning the modelling of cryptography primitives as SIBs and their organization in domain-specific SIB palettes. We also sketch how to use multifaceted taxonomies to provide compact yet expressive classifications, amounting to a semantic description of the security domain. We address in the issue of workflows by using models that ease the expression, analysis, control, and formal verification of inter- and intra-model control and data flow, though the adoption of the XMDD approach implemented in the DIME integrated modelling environment. A brief description of a home banking application sketches how in reality many of these security mechanisms need to work together in a safe and secure orchestration.**

I. Introduction

Cryptographic and security systems are the basis for guaranteeing properties like confidentiality, privacy, authentication and data integrity in several critical aspects of our society, like communications, banking, commerce, government, defence, and national security. These systems rely on the use of security primitives that allow the implementation of such properties.

Security primitives are low-level cryptographic hardware modules and algorithms used to guarantee security for computer systems. Even though security systems are widely distributed and used in all digital applications, their actual implementation remains still a challenging task, since the designer has to meet applications constrains (in terms of speed, throughput, costs) and at the same time to cope with the hectic market rules that do not give enough time for full validation and testing of such products.

Veracode an application security testing company, surveyed in 2015 the difficulties software developers declare in implementing cryptographic algorithms [1]. Analyzing the code Veracode customers submitted to its platform over an 18 month period along the OWASP[2] top10 vulnerability categories, cryptographic issues of application across all industries ranked second behind overall code quality, with information leakage as overall third. As shown in Fig. 1, these three categories were found to be the top 3 in 5 (Financial Services, Healthcare, Retail and Hospitality, Technology, and Others) of the 7 verticals surveyed, while in Government cryptographic issues ranked 5th with 51% prevalence and 4th in Manufacturing with 45% prevalence.[1]

Vulnerability	Financial Services	Government	Healthcare	Manufacturing	Retail & Hospitality	Technology	Other	Rank
Code Quality	65%	70%	80%	56%	68%	70%	65%	1
Cryptographic Issues	60%	66%	61%	51%	63%	62%	59%	2
Information Leakage	58%	62%	60%	49%	55%	62%	53%	3

Fig. 1. Top 3 vulnerability categories by industry vertical - from [1] p.11

Along this analysis, we see that the top 3 issues concern code in general, cryptography mastery, and information flows.

In our SEcube^{TM} platform, we propose to

- reap the benefits of model driven design, and lift the issue of application design and management mastery from the exclusive code level to a model driven approach, as described in [3] and applied in [4],
- provide a set of predefined, high cryptography primitives, as described exemplarily in Section II, making them available for use within the modelling environment as a domain specific palette of primitives, as described in Section III, and subsequently as a library of generated code,
- profile these functionalities according to relevant description facets and make them available in a semantically accessible model driven fashion, as described in Section IV,
- address in the issue of flows by using models that ease the expression, analysis, control, and formal verification of inter- and intra-model control and data flow, though the adoption of the XMDD approach [5] implemented in the DIME integrated modelling environment [3], [6],
- support the domain-specific and (security) aspect analysis via properties that express the guarantees needed to enforce well behaved, secure execution. These properties, as discussed in [4], can often be enforced at design time,

[1] As shown in the same report, with an overall flaw density of 352 flaws/MB and a very high or high severity density of 54 flaws/MB, Manufacturing appears to be the sector in most need of improvement, in comparison to Technology (83 resp. 29 flaws/MB) and Government (62 resp. 7 flaws/MB)

ISBN: 1-60132-445-6, CSREA Press ©

on the models, this way helping to reduce the amount of costly and slow code-level analysis or testing otherwise unavoidable.

In the following, we concentrate on a few exemplary security primitives (Section II), then sketch how a security primitives palette looks like in DIME (Section III), and in its semantic description by means of layered taxonomies (Section IV). We then describe how model driven design of Security Processes is made possible in DIME thanks to its support of variability, aspect orientation, and loose programming (Section V), and lightly analyze the interplay of different techniques for the security of a home banking example (Section VI).

II. Security Primitives

Security primitives are low-level functions, modules, and algorithms that allow adding security capabilities to digital devices, protocols, and applications. In the following we briefly describe the most important primitives: encryption algorithms (both symmetric and asymmetric), Physically Unclonable Functions, and True Random Number Generators. For a systematic introduction to applied cryptography refer to [7].

A. Encryption Algorithms

Encryption algorithms are used when is needed to preserve the confidentiality and the privacy of data when this is stored or transferred. Security is ensured altering the information message to be exchanged or saved. The information to be processed, the *plaintext*, passes through a series of mathematical operations (e.g., transposition, substitutions, etc) depending on the encryption algorithm, in order to be encrypted. The result obtained is an encoded information, the *cyphertext*, where the data is no longer clear and only authorized parties are able to read it. During both encoding and decoding processes, the cryptographic key is used as a parameter to specify the transformations of the input data. Moreover, the length of key has to be long "enough" so that exhaustive malicious attacks become unfeasible or too costly.

Depending on the key, encryption algorithms can be classified in symmetric and asymmetric. In symmetric encryption algorithms the key is common to both encryption and decryption stage. This requirements is one of the main drawbacks. Examples of symmetric encryption algorithms are: Blowfish, DES (Data Encryption Standard), 3DES (Triple DES), AES (Advanced Encryption Standard, also known as Rijndael), IDEA (International Data Encryption Algorithm), RC5. Conversely, for asymmetric cryptography (or public-key cryptography) a pair of keys is used to encrypt/decrypt data. Every user owns a pair of key, one public and the other one private. The message is encrypted with the public key, but decrypted with the private one. Public-key algorithms include RSA (Rivest - Shamir - Adleman), DSA (Digital Signature Algorithm), Diffie-Hellman, and ECC (Elliptic Curves Cryptography).

B. Physically Unclonable Functions

Physically Unclonable Functions (PUFs) exploit intrinsic manufacturing variability existing during the fabrication process of integrated circuits in order to generate a signature, unique to each single device. PUFs are a replacement of existing solutions based on Read-Only Memories (programmed at manufacturing time) or non-volatile One-Time Programmable memories. These existing solutions are shown to be vulnerable to reverse-engineering attacks and thus they cannot guarantee high security.

The produced signature must be unique from device to device, unclonable, and, for a same device, it must be robust with respect to ageing and environmental variations (reproducible). The adopted underlying mechanism is a challenge-response generator. A PUF needs an input, i.e., the challenge, to produce an output, i.e., the response. The challenge-response pairs (CRPs) set must be unique for a single device.

PUFs can be used either to generate secret keys used in encryption algorithms, or to generate a set of CRPs used to authenticate the physical device itself. As for the authentication, the PUF is first queried in order to obtain a significant subset of CRPs to save in a secure server. Once deployed, the PUF will be queried with that set of challenges. If the signatures generated by the device are equal to those stored in the server, the device is authenticated. PUFs with a significant amount of CRPs, such that is unfeasible for an attacker to exhaustively stimulate the PUF with all the allowed challenges, are classified as *strong* PUFs. On the other side, PUFs with small amount of CRPs are defined *weak*. Furthermore, some PUFs are designed to retrieve only one response, as a single signature. Typically they are adopted for key generation and storing.

One of the most investigated solutions uses SRAMs, since they provide high security (i.e., high inter-chip variation) and high stability (i.e., low intra-chip variation). Commercial devices and state-of-the-art studies exist for current SRAM CMOS technologies.

In the context of our work, we want to provide SEcube users with an easy way to access the PUF, without the need of understanding the underlying electrical and intrinsic physical mechanisms exploited for the PUF to work. A weak PUF will be seen as a constant from the programmer, whose value will be always the same for a same device, and always different for different devices. On the other side, a strong PUF will be seen as an array of weak PUF, one for each challenge. The parameters of the PUF will be the size of the generate response and the number of possible challenges.

C. True Random Number Generators

True Random Number Generators (TRNGs) are used to generate random numbers from a physical process, rather than a fixed algorithm of a predictable computer program. They are implemented by taking advantage from a physical process, like thermal noise or any other quantum phenomena and are expected to generate random bits with very high entropy and zero correlation. An on-chip TRNG design should

occupy small area, give high bit rate, and have low power consumption, while assuring un-biased bit streams with high entropy per bit and low (no) correlation among them. In our work, a TRNG will be seen as a function generating a random number each time it is called. The parameters of the TRNG will be the size of the generate number and the maximum obtainable throughput.

III. Security Primitives Palette in DIME

Security primitives like those described in the previous section could be employed in different domains of interest at many levels of abstraction.

Due to the large quantity of primitives available, an efficient organization is required in order to easily locate and use them in the most appropriate way. As over time such collections grow large and more diverse, their organization must be easily understandable to different stakeholders: those how use them, manage them, and maintain them. Also here, the coarse granular model driven approach adopted in the DIME [] integrated modeling environment helps maintain the essential information well represented at the surface, while hiding the internal organization and more detailed traits of its description and implementation, that can be accessed at need on demand.

At the level of individual primitives, every security primitive becomes an atomic domain specific SIB (*Service Independent Building Block*) of the `Security` domain. The SIBs for a given domain offer what the domain experts consider to be n appropriate and useful basic service, able to satisfy a specific function.

As briefly introduced in [8] and illustrated in [3], DIME has a number of model types and views that collectively form a multifaceted yet coherent description of the system under design. The DIME Diagram Editor provides the canvas to draw the various graphical models, and additionally it provides also the palettes with basic model components that are in use for the of the model currently open and under design. These palettes can be

- **specific to the subject matter domain relevant for the specific case study**, like diabetic outpatient treatment as in [9], bioinformatics as in [10], [11], geo-information systems like in [12], but also home banking operations as in the case study of Sect.VI, or computer vision and robotics as in [13], but also
- other **cross-sectoral palettes** of functionalities needed, e.g., for security or communication, that are themselves domains, but find their use largely in combination with and embedded within (any) application that, like the above mentioned ones, would be primarily classified in another domain.

Each SIB can be dragged on to the canvas and linked with others in order to *"draw"* the workflow implementing a larger service or process. Therefore, for each SIB, the essential information at the model level concerns its correct use and embedding in (potentially any) context. At a minimum, as for APIs and Services in a service-oriented paradigm, this spans their correct embedding inside

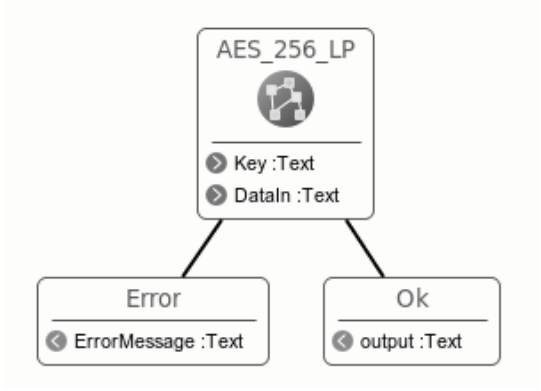


Fig. 2. SIB `AES-256` in DIME: control flow and data flow

- **Data models**, that cover the design of domain models based on common concepts like classes, attributes, and uni- or bi-directional relations between elements, and
- **Process models**, whose types in DIME span the core business logic, data retrieval (search queries) as well as dynamic access control (security guards, particularly interesting for the SEcube^{TM} platform).

Considering for example the SIB implementing the encryption function of AES shown in Figure 2, we see that

- its **Icon** denotes that it is itself a hierarchic model. Accordingly, one finds it listed in the DIME Model viewer, one could open the corresponding model and inspect the Service Logic Graph of its logical internal flow,
- its **API** expects as inputs a `Key` of type `Text` and `DataIn`, input data of type `Text` as well, and its outputs are an `Output` and an `ErrorMessage`, both of type `Text`.

In terms of a normal API description, as found in Architectural DSLs, in WSDL description of services, in the SCA SOA standard, and in the widely practiced component based design in software engineering, this would be all the information available on this component. It is sufficient from an architectural point of view to describe its execution-independent I/O potential, i.e. its static "pluggability", but it does not describe its behaviour, essential to use it properly.

In DIME, however, we model also the contro flow:

- the **control flow** foresees two outgoing branches: if no issue arose upon execution, the `Output` data is produced and the execution continues with the SIB that is connected to the outgoing branch labelled `Ok`. If some error occurred, an `ErrorMessage` is prepared and sent to the (exception handling) SIB connected with the `Error` branch.
- all these **labels** are viewed by the checking algorithms, that check both completeness (e.g, no dangling branches in a Service Logic Graph), (type) correctness when composing SIBs, and the correctness of the logical flow.

A large set of the properties expressed in Sect. 4 of [4] in fact

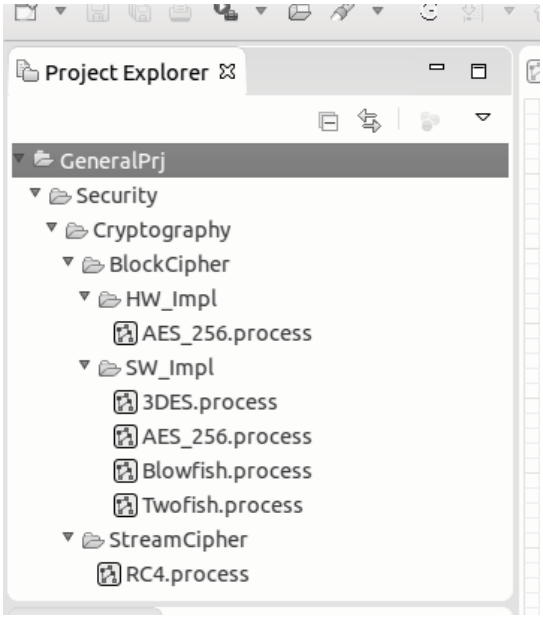


Fig. 3. Excerpt of the Security SIB Services in DIME

concerns morphologic and control flow matters, and needs the information contained in these labels.

IV. Semantic description of the Security DSL

The AES algorithm belongs to the `Cryptography` subdomain of the `Security` domain and it is a `Block Cypher`. In DIME, a user can find it under these same headers browsing the directory path as shown in Figure 3. To find it easily, one needs either a consolidated knowledge of security, or a search mechanism. However, we see that `AES_256` appears twice: once as a hardware and once as software implementation. All the security primitives seen in the previous section, and all the SEcube and the libraries relative to application domains, can be organized in taxonomies, i.e., concept based classifications with domain specific categories, whereby categories lower in the taxonomy are specializations of the higher ones (formally this is an **is_a** relation). The SIBs are leaves of such a taxonomy. The taxonomies we use are multifaceted: a tree might not be flexible enough, since some SIBs might belong simultaneously to multiple categories on different branches. In our an example, the encryption primitive for `AES_256` might be available in both hardware and software implementations, thus belonging to both `HW_Impl` and `SW_Impl` categories.

While specialists can leverage their knowledge of the domain and its terminology to find what they are looking for, casual user are easily lost. For them, not knowing the domain vocabulary, it is important to support a declarative, query-like approach. Given a taxonomy, these users can navigate through the linked concepts, and query this structure with questions like

Which AES implementations are not Hardware based?

Referring to the strongly simplified taxonomy in Figure 4, this query translates in a logical expression

$$256-\texttt{bit} \wedge \neg \texttt{Hardware}$$

that return the sets of SIBs that satisfy that property.

For our applications in the DIME based models, a suitably expressive classification language is achieved through taxonomies that are DAGs (directed acyclic graphs). A taxonomy is a special case of an ontology, more precisely a tree, like in the phylogenetic trees of species in biology, or if multifaceted, like in our case, a DAG. Fig.4 illustrates small fragments of the `Security` and `Implementation` facets.

More generally, a domain ontology provides a shared vocabulary, which can be used to model a particular domain, i.e., the type of objects and/or concepts that exist, and their properties and relations. The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application.

The creation of ontologies is a large collective effort: ontologies are a precious good that codifies the knowledge and the language of a community of practice. The knowledge expresses there is typically layered: upper ontologies define general terms of the domain. In our simplified Fig. 4, the top level terms are those likely to belong to an upper ontology. The bulk of the general domain specific knowledge is defined in middle ontologies, in the hands of *domain experts* in the particular field of interest, who define its stable vocabulary. Although formally defined ontologies and automatic reasoning are still not widely adopted, a body of de-facto classifications, mostly in tree or matrix form has crystallized in most communities. For example, Fig. 5 reproduces an informal classification tree for System properties that distinguishes among Functional and Non-functional ones, listing some at the bottom. We see here one of the weaknesses of informally expressed domain knowledge: where exactly is each of the 4 lists connected to? some have titles (seeming like a lower ontology connection, eg, the ”-ilities” header), some not, and certainly this depiction is not machine readable, thus cannot be queried.

While upper ontologies should in the long term change rarely, middle and lower ontologies are frequently updated by concept and relation refinement. In our case, the middle ontology includes concepts like `Block Cypher` and `Stream Cypher`, that are likely ignored outside the security community. Lower ontologies are closer to the instances, that evolve rapidly and are more dynamic, following the evolution of the technology, platforms, and needs.

Verification and synthesis methods like those described in the next Section depend on the knowledge of properties of the things they deal with, thus they need to refer to machine readable descriptions of such things, codifications of the domain they belong to in terms of concepts and relations among concepts and with the things, in order to navigate and query these concept/relation networks.

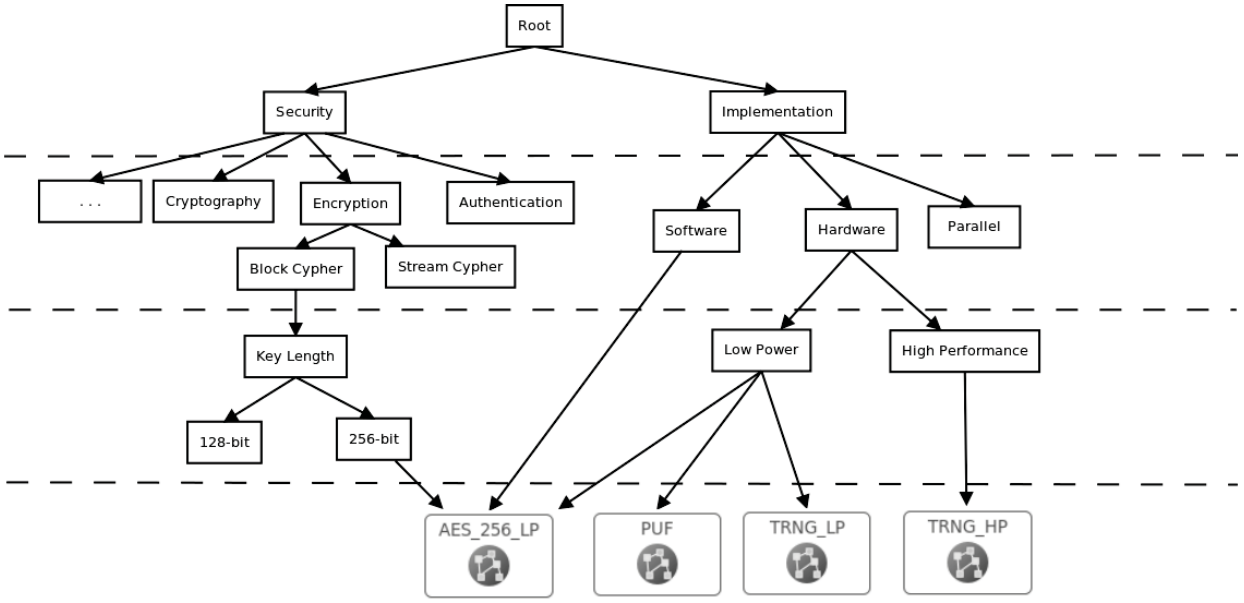


Fig. 4. Fragment of a simplified taxonomy in the security domain

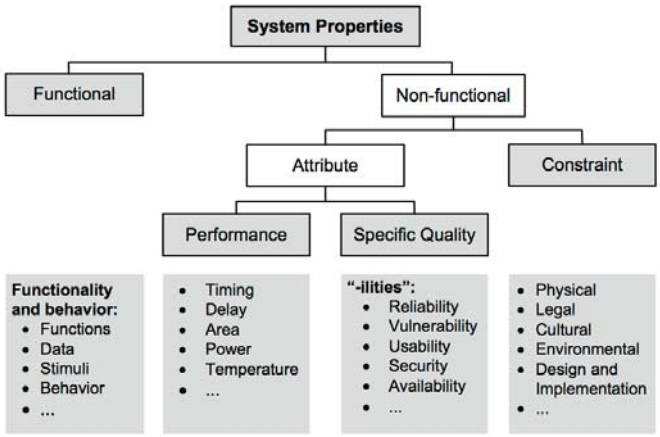


Fig. 5. Taxonomy of System Properties, from [14]

V. Towards Model Driven Design of Security Processes in DIME

As described in Section II, encryption algorithms can be classified as symmetrical and asymmetrical according to the encryption key nature. There are several algorithms created on the base of mathematical properties (e.g. big numbers multiplication, exponentiation, etc.) and entropy generation techniques (e.g. mix column, permutation tables, etc.). In spite of these significant internal differences, it is possible to define a common execution process and an abstract interface for all the algorithms in the same class.

For example, symmetrical algorithms can be managed through the same programming interface, summarized with the following functions: `Initialize`, `SetIV` (meaning Set Initial Vector), `Update`, `Finalize`.

- The `Initialize` function specifies the algorithm direction (e.g. encryption mode, decryption mode), the feedback mode (e.g. ECB, CBC, CFB, etc.) and starts the key expansion procedure.
- The `SetIV` function is only used in a feedback mode and it sets the initial vector. In principle the SetIV function may be included in the Initialize phase. Nevertheless, since the initial vector could be changed at any time and the key expansion procedure (which is usually time consuming) is not required, the SetIV functions can be isolated for a more effective usage, and the `Initialize` function becomes itself a process, that internally uses `SetIV` (cf the Usex discussion in [8]).
- The `Update` function just processes a data buffer (n blocks in case of block-cipher algorithms or n-bytes in case of stream-cipher algorithms) according to the algorithm configuration (direction, feedback mode, etc.).
- Last, the `Finalize` function frees the internal implementation structures, especially in implementations able to manage multiple concurrent sessions.

This is a case of set up and usage process that is *parameterized* in the concrete algorithm i.e. independent of which one among the set of symmetrical algorithms currently available is chosen. This is interesting, in that it allows to express in our platform the *shape of this interaction* independently of the instance of behaviour (i.e. which concrete run is executed) and of architecture (i.e. which concrete components are plugged in). In terms of the philosophy explained in [15], [16], this is a case of *horizontal and vertical looseness*.

Vertically, we employ hierarchical modeling as an aspect-oriented mechanism for specifying variability. In this case, we will use placeholder SIBs at all the variation points, that specify the characterization of the suitable instances in

terms of the currently valid taxonomy concepts. This way, any SIB (elementary or not) that satisfies the properties of the variation point is an eligible instantiation, and thus a correct variant. This needs to be complemented by *constraint-guarded variability modeling*: Model checking needs to be applied in order to establish the global consistency of the product variants, which are typically built by manual specification of variations points. In this case, even if any of the symmetrical algorithms is eligible, once we have instantiated our choice concrete algorithm, it needs to remain the same along the entire protocol. This ”sameness” can be expressed as an additional constraint and checked efficiently.

Horizontally, this process is actually spread at several locations inside the larger process of the application, which is the context (or host) of the embedded security process. Therefore, its description and modelling in DIME is de facto a template of the shape, whereby most of the control flow will not be contiguous, but be interrupted by intermittent portions of the application process. For example, it is to be expected that the application will have a prologue before reaching the `Initialize` SIB, then there will be one or more occurrences of the `Update` SIB, either directly after `Initialize` or intercalated in the application workflow, and that even the `Finalize` SIB may not be the last SIB executed by the application. Here, the loose programming paradigm of [] can help. Equivalent to declarative properties, these templates can be described in terms of ”must-”successors that are in all instances the concrete next SIBs, and ”eventually” successors, that can occur along any given path after a number of other SIBs. For this kind of templates it is possible to do manual refinement followed by checks, but also to resort to constraint-driven variability modeling using (LTL) synthesis technology [17], [18], [16], to fully automatically generate workflows that satisfy all given shape constraints.

VI. Case Study: Home Banking

Nowadays, home banking is one of the most widely used web-based secure services. Several security mechanisms protect both the customer and the bank during the various processes. Although mechanisms and strategies may vary according to the banking institutions, the security primitives provided by the SEcube platform and their correspondent modelling can be combined to cover any occurring security scenario concerning the bank, the user, and the operations to be carried on with the bank accounts.

The right Bank. For example, when the customers initiate a web connection to the bank website to start a home banking session, the first security procedure is implemented through the HTTPS protocol: it aims to authenticate the bank website and create an encrypted communication channel.

This process is usually implemented by digital certificates: the bank website exposes its identity and public key through a certificate delivered and signed by a certification authority recognized by the customer web browser. This process guarantees that the user is not connected to a fake bank web site.

In terms of security primitives, the certificates management usually requires algorithms like RSA and DSA, which are fully supported and modeled in the SEcube platform.

Once the web bank authentication is successfully performed, a secure HTTPS channel is established using symmetrical encryption algorithms, such as AES256, and the user must authenticate itself.

The right User. There are many ways for the users to be authenticated. However, all the methodologies provided by the banks are based on a multi-factor authentication process, which requires more than one factor (e.g. username, password, one-time password, token authentication, etc.) to prove the user's identity and authenticity. Again, in this case the low level security primitives provided by our platform can be combined to implement the higher level mechanisms. For example, the one-time password can be implemented as an encryption algorithm evolution started from a basic key and plaintext (also called seed) which are in common between the user token and the bank server.

Once both the parties are authenticated, there are several ways to create symmetrical session keys to protect the communication channel. In most of the cases the symmetrical keys are derived by the random challenges used in the mutual-authentication process. Sometimes the session keys are created and exchanged using specific asymmetrical algorithms like DH (Diffie Hellman). A very few times, the session keys are generated from a pre-shared master key. In any case, all the techniques described above (and many others) can be implemented by a combination of the security primitives provided by the platform.

The right Operation. After the mutual authentication is performed and the secure channel is established, the secure service is in place and the sensitive information can be encrypted and signed. In order to prevent reply attacks, the communication protocols usually provide counters which are automatically incremented at any packet transmission.

According to the banking policies, the signature process can be performed at any transmitted packet or just for specific operations, such as money transactions, authorizations, etc. In any case the algorithms used for signature belong to the asymmetrical class: RSA, DSA, Elliptic Curves, etc.

More security behind the scenes. Sometimes, within the same working session, the cryptographic keys can be renewed in a transparent way (without any action on the user side). In any case the security primitives are still the same independently of the mechanisms and strategies implemented by the specific home banking services.

Wrapping up properly. Finally, when the working session is terminated, the secure channel is closed, the session keys are deleted and all the encrypted/signed packets used in the previous session are invalidated against possible reply attacks attempts.

VII. Conclusion

In this paper we have shown how the consequently modular hardware and software architecture of the SEcube platform

paired with the DIME integrated modelling environment and semantic domain description techniques can render security more easily understandable and reusable at large scale, also by non-experts in security, With an adequate formal representation of the sematic domain and its properties and relations, and with better support for property expression and enforcement, we are confident that the SEcube environment as a whole can significantly contribute to a wider adoption and higher quality implementation of security also in vertical domains where so far it is still difficult to master. Referring again to the Veracrypt report [1], their reporting that the prevalence of Cryptographic issues ranged from 80% of the uploaded applications analyzed provenient from the Healthcare sector, to 63% in Retail and Hospitality 62% in Technology, 60% in the Financial Sector, 51% in Government, down to 45% in Manufacturing, is discomforting. Healthcare and Financial Sector, as highly regulated industries (like Transportation, that was not surveyed as a category) are mandated to achieve high rates of security. One of the observations of the report concerns the programming language of the analyzed over 200.000 projects. The vast majority of the application was written in .NET or Java, and the choice maes a difference. "*Where some languages and programming models completely eliminate some security issues (for instance, buffer management issues common in C/C++ are completely eliminated in Java or .NET), often the choice of programming language is influenced by factors other than security*. This indicates that there are benefits in raising the level of abstraction ant which programmers work, providing a language discipline, infrastructure, and tools that take care in a preventive way that certain issued do not arise. Moving to models instead of code for secure application design can be the next generation of abstraction that helps to scale pervasively.

ACKNOWLEDGMENT

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

We also thank Antonio Varriale (Blu5 Labs) for his valuable input to this work.

REFERENCES

[1] Chris Wysopal. State of software security - focus on application development (2015). https://www.veracode.com/.

[2] The Open Web Application Security Project. https://www.owasp.org/index.php/Main_Page.

[3] Steve Boßelmann, Johannes Neubauer, Stefan Naujokat, and Bernhard Steffen. Model driven design of secure high assurance systems: an introduction to the open platform from the user perspective. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[4] Giuseppe Air Farulla, Marco Indaco, Axel Legay, and Tiziana Margaria. Model driven design of secure properties for vision-based applications: A case study. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[5] Tiziana Margaria and Bernhard Steffen. Agile IT: Thinking in User-Centric Models. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation*, volume 17 of *Communications in Computer and Information Science*, pages 490–502. Springer Berlin / Heidelberg, 2009.

[6] Stefan Naujokat, Michael Lybecait, Dawid Kopetzki, and Bernhard Steffen. Cinco: A simplicity-driven approach to full generation of domain-specific graphical modeling tools. *Int. Journal on Software Tools for Technology Transfer (STTT), Springer Verlag, (to appear)*, 2016.

[7] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Publishing, 2010.

[8] Antonio Varriale, Giorgio Di Natale, Paolo Prinetto, Bernhard Steffen, and Tiziana Margaria. Secube^{TM}: An open security platform: General approach and strategies. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[9] Tiziana Margaria, Steve Boßelmann, and Bertold Kujath. Simple modeling of executable role-based workflows: An application in the healthcare domain. *J. Integrated Design & Process Science*, 17(3):25–45, 2013.

[10] Anna-Lena Lamprecht and Tiziana Margaria. Scientific Workflows and XMDD. In *Process Design for Natural Scientists: An Agile Model-Driven Approach*, volume 500 of *CCIS*. Springer Berlin Heidelberg, 2014.

[11] Anna-Lena Lamprecht, Tiziana Margaria, and Bernhard Steffen. Seven variations of an alignment workflow - an illustration of agile process design and management in bio-jeti. In *Bioinformatics Research and Applications, Fourth International Symposium, ISBRA 2008, Atlanta, GA, USA, May 6-9, 2008. Proceedings*, pages 445–456, 2008.

[12] Samih Al-Areqi, Steffen Kriewald, Anna-Lena Lamprecht, Dominik Reusser, Markus Wrobel, and Tiziana Margaria. Towards a flexible assessment of climate impacts: The example of agile workflows for the ci: grasp platform. In *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications - 6th International Symposium, ISoLA 2014, Imperial, Corfu, Greece, October 8-11, 2014, Proceedings, Part II*, pages 420–435, 2014.

[13] Giorgio Di Natale, Alberto Carelli, Pascal Trotta, and Tiziana Margaria. Model driven design of crypto primitives and processes. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[14] Nadereh Hatami Mazinani. Multi-level analysis of non-functional properties. PhD Thesis, Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart, 2014.

[15] Anna-Lena Lamprecht, Stefan Naujokat, and Ina Schaefer. Variability management beyond feature models. *IEEE Computer*, 46(11):48–54, 2013.

[16] Sven Jörges, Anna-Lena Lamprecht, Tiziana Margaria, Ina Schaefer, and Bernhard Steffen. A constraint-based variability modeling framework. *International Journal on Software Tools for Technology Transfer*, 14(5):511–530, 2012.

[17] Bernhard Steffen, Tiziana Margaria, and Burkhard Freitag. Module configuration by minimal model construction. In *Technical Report - MIP Universitaet Passau, Faku—taet fuer Mathematik und Informatik*. Citeseer, 1993.

[18] Stefan Naujokat, Anna-Lena Lamprecht, and Bernhard Steffen. Loose programming with PROPHETS. In *Fundamental Approaches to Software Engineering - 15th International Conference, FASE 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, pages 94–98, 2012.

Model driven design of secure properties for vision-based applications: A case study

Giuseppe Airò Farulla*, Marco Indaco
Looqui Srl and Politecnico di Torino, Turin, Italy
giuseppe@looquilabs.com, marco@looquilabs.com

Axel Legay
INRIA, Rennes, France
axel.legay@inria.fr

Tiziana Margaria
Univ. Limerick and Lero, Limerick, Ireland
tiziana.margaria@lero.ie

∗ corresponding author

***Abstract*—In this paper we discuss an approach to overcome difficulties and gaps which are typically encountered when dealing with security-oriented model-driven approaches. In particular, we state that state-of-the-art MDS approaches are not suitable for modern companies and industry in general, and address security only at a late stage of development, often causing big delays and reengineering costs due to extensive reworks. Instead, we propose to adopt in the SEcube platform an OTA-based XMDD approach to integrate security ab-initio. In addition, since our approach is based on a set of reusable SIBs organized within dedicated palettes in DIME, we decouple the issue of guaranteeing that the SIBs are correct and secure from the issue of analyzing the applications, which can be greatly simplified by knowing the characterization of each SIB in advance. We apply our approach to the concrete realm of computer vision steering robotics, present the safety and security properties elicited on the specific case study, and discuss the ways they can be enforced.**

I. Introduction

Continuous Model-Driven Engineering (CMDE) [1] is a software and system development methodology which focuses on creating and exploiting domain-specific models, which are conceptual models of all the artefacts, concepts, actions, and properties related to a specific system under development. Hence it aims at creating abstract and possibly reusable representations of the knowledge and activities that govern a particular application domain, rather than focusing on the computing (i.e., algorithmic) concepts.

The CMDE approach is meant to increase productivity along the entire life-cycle by maximizing compatibility between systems (via reuse of standardized models), simplifying the design process (via models of recurring design patterns in the application domain), and promoting communication between individuals and teams working on the system (via standardization of the terminology and the best practices used in the application domain).

Model-Driven Architecture (MDA) is a model-based software design approach for the development of software systems. Traditional Model Driven Development (MDD) of software, as promoted e.g., by the OMG, shifts the attention and the design activities from the programming level to the modeling level, but still remains in the IT realm. Even at the platform independent (PIM) level, the typical UML-based MDD and MDA-approaches provide varieties of model structures that focus on different technical issues/aspects, that have separated life-cycles. Mastering this wealth is a special art requiring both IT knowledge and a good ability of dealing with abstraction.

The MDD paradigm relies on the use of Domain-Specific Modeling Languages that incorporate as core objects (also called "first-class citizens") elements of the domain being modeled and their relationships, and model transformations that transform the models into platform-specific artefacts, such as code [2].

In this paper we focus on security aspects in CMDE, concerning modeling of secure properties in the concrete realm of Computer Vision (CV) and vision-based algorithms, used in many fields of modern IT. We also present and discuss a practical case study.

Model-Driven Security (MDS) has for more than a decade proposed methodologies for supporting the development of secure systems, or the development of systems leveraging on secure properties and actions to protect mission-critical tasks. Yet, according to studies and state-of-the-art papers such as [3], there is still a big gap between the current practice and what is need to make MDS more easily applicable and adoptable by companies and industry. Most current MDS approaches do not extensively deal with multiple security concerns but rather focus on a specific one, i.e., authorization or authentication.

Security patterns, based on domain-independent, time-proven security knowledge and expertise, could be considered as reusable security bricks upon which sound and secure systems can be built. They are however not actually applied as much as they could be. This is mainly due to two reasons:

- developers have problems in selecting them and applying the right pattern in the right places, especially at the design phase, because they are not used to this programming modality, and
- although the framework enables powerful and almost limitless security federation in principle, it requires users to have deep knowledge of security, and an understanding of the security infrastructures.

To become truly pervasive, as required in today's interconnected and des-intermediated world, security should be unified with the software engineering process, and thus security engineering is of great importance [4].

ISBN: 1-60132-445-6, CSREA Press ©

Unfortunately, often security is considered as an afterthought in most actual development, and treated as an add-on after the functional requirements are implemented. While it is often considered only during the implementation or deployment phases, it is well known that finding defects downstream greatly increases the costs of removal and repair.

The One Thing Approach [5] for model construction and management, and the eXtreme Model Driven Development (XMDD) [6], [7] as collaborative software engineering process can help considerably lower the barrier to built-in security form inception.

II. Modeling Security with the One Thing Approach

Developing systems with the eXtreme Model-Driven Development (XMDD) paradigm [6], [7] involves the user/application expert continuously throughout the whole systems' life-cycle, according to a user-in-the-loop and expert-in-the-loop philosophy [8]. It is model-driven because it is based on the One-Thing Approach (OTA) [5], [9], which works by successively enriching and refining single artefact that is a rich multi-aspect and multi-faceted model. We use the DIME [10] tool, a Cinco [11] product that is adequate for modeling the aspects of concern here. As illustrated in [10], it is possible to model an application-level security aspect through Role Based Access Control (RBAC). In addition, as shown in [12] and [13], within DIME it is possible to use Service Independent Building Blocks (SIBs) that provide a service-oriented library of basic security communication for Data at rest and Data in motion, as well as a choice of higher level primitives and protocols that can be embedded in the application under design [14]. The entire SEcube^{TM} platform [15] is therefore prepared to support users in adding security aspects to the model, by leveraging predefined abstract security primitives, which they might theoretically not even know nor understand in detail.

In this paradigm, our application models are at the center of the design activity and the first class entities of the global system design process. In this approach:

- domain specific libraries are established at the model level: our building blocks are (elementary) models rather than software components;
- systems are specified by model assembly. Here we use orchestration, hierarchy, and configuration as composition techniques;
- knowledge and requirements are expressed by means of properties, via constraints that are formulated in an automatically verifiable fashion. Actually, some of the constraints happen to be domain-independent, and to be already taken care of at design time of DIME;
- security is layered inside SIBs, at the SLG level, and at the global level including the run-time environment. We will see this briefly described on the case study;
- system changes (e.g., upgrades, customer-specific adaptations, new versions) occur only, or at least primarily, at the model level, with a subsequent global re-verification, and re-compilation (or re-synthesis, in the future);
- optimizations are kept distinct from design issues, in order to maintain information on the structure and the design decisions independently of the considerations that lead to a particular optimized implementation.

For these reasons, DIME includes by design the support of properties and model manipulations that are foundational for the OTA-based XMDD. DIME focuses on application experts, who are typically non programmers, and its versatility is one of its key characteristics.

Although there has been a lot of research on security issues concerning technologies, we want to address business-level security intent at a level that is easy to understand even for business users.

III. Integration of security aspects

The current state-of-the-art in developing security-critical software and systems in practice is far from being satisfactory. New security vulnerabilities are discovered on an almost daily basis. Integration of security into the overall development process is problematic and suffers from two gaps [16]. First, it is possible to identify a gap as security models and system design models are typically disjoint and expressed in different ways. Second, although security requirements and threats are often considered during the early development phases (requirements analysis), there is another gap with security mechanisms which are later employed and implemented in the final development phases (system integration and validation). As a result, security is typically integrated into systems in a post-hoc manner, this way degrading the security and maintainability of the resulting systems.

To address this problem, a significant amount of work over the last decades has provided model-based development approaches [16]. State-of-the-art approaches which aim to raise the trustworthiness and intrinsic security of mission-critical systems are typically based on the Unified Modeling Language (UML) [17], consisting of a collection of formalisms intended to collectively provide a standard way to define and visualize the architecture and behavior of a system. However, we believe that UML is neither formal nor intuitive enough to be easily understood by application experts, and advocate the need of a XMDD-based approach for security to gain better formality, consistency, and at the same time full notation-independency.

The aim is to integrate security aspects ab initio, from the high-level definition of processes all the way through the whole development of embedded/cyber-physical systems. This will enable true system-immanent security modeling. By integrating security and XMDD, it is possible to model and generate security aware applications that only present options to the user that are consistent with the formalized security policy. Users can be also provided with short checklists (in reality palettes of constraints) summarizing established rules of prudent secure engineering. Our approach towards model-based security engineering combines the following strategies:

- definition of security primitives;
- automatic analysis of models against security requirements;
- automatic generation of code (or tests) from models.

Before presenting our approach it is interesting to define what could be an ideal primitive, which varies in its traits between different application domains. In our domain, security primitives should be composed of a hierarchy of controllers and a data path. We are interested in implementing through such a primitive security controllers and to enforce integrity and privacy within our resources, intended both as data and as processes. The main idea is to define security modeling languages to form a general-purpose extensible DSL, in the sense that they leave open the nature of the protected resources, that belong to some vertical application domain, and the nature of the rules of the game.

Starting from this consideration, our SIBs and Service Logic Graph (SLGs, graphs representing the control flow of an application) are created by modeling primitives and generation rules for integrating security into the development process, i.e., including the resources the security primitive protects. Graphically, they will be visualized within colored boxes, indicating that they are executed under surveillance of the security controller.

Once designed, the SIBs managing critical accesses or operations will be easily reusable in other applications of the same, or similar, domain, to create new secure applications. Secure SIBs will be easily reusable also by developers not specialized in secure systems, as they do not need to be aware of the enforced security properties: they can simply include these SIBs in their designs, and rely on the safety controller for guaranteeing the security of the application models. We are designing a methodology as a general schema that allows designers to specify system models along with their security requirements and then use DIME to automatically generate the actual system architectures from the own models plus the security aspect once available as sketched in [13], including complete, configured access control infrastructures. This way, we pave the way towards an integrated system which closes the above mentioned gaps: the gap between security models and system design models, and the gap between design and implementation. We accomplish this by a model-driven development process where security is explicitly integrated in all the phases of the design process. To make a high-level SLG security aware, we combine it with low-level SIBs implementing security primitives, intended for guaranteeing privacy, integrity and authenticity of all its composing SIBs (or of the most critical ones). The security controller checks at design time the correctness against static rules about, e.g., allowed roles, users, and permissions, and at run-time evaluates dynamic code to cover monitoring and run-time protection aspects.

In our approach, we have defined three basic levels of security for the domains we consider.

No security The first level, representing the lowest level in our classification, is reserved for SIBs which do not require to be protected, either because they do not perform mission-critical operations, or because they do not interact with other machines nor hardware, so there is no communication to secure. Still, an attacker could exploit vulnerabilities in the code of these SIBs, but could not gain advantages to compromise the execution of the whole SLG. SIBs belonging to this level of classification do not have a graphical marker.

Medium security: secure SIBs The second level in our classification is reserved for SIBs which are still not mission critical, although they perform inter-machines actions, for instance exchanging messages over the network, thus requiring the establishment of a secure and confidential transmission line, and the verification that the user executing the SIB has the right permission to do so. Secure SIBs belonging to this level of classification are graphically identified by a surrounding yellow box in our DIME models. They can use the Data at rest/Data in motion libraries of [12], or the design and protection mechanisms introduced in [10] and [13].

High security: critical SIBs The third level, finally, is reserved for mission critical SIBs. These SIBs execute operations which, if changed or disturbed from a malicious attacker, could compromise the execution of the whole SLG. In addition, their operations require both inter-machines communication and communication to/from hardware devices, that need special protection to ensure that attacker cannot change or interfere with them. For instance, these SIBs check both the user executing them, to evaluate if it belongs to a group of authorized subjects thus having the right permissions, and the hardware, verifying its integrity and the device characteristics (e.g., ID, provider) against a list of authorized devices. Critical SIBs belonging to this level of classification are graphically identified by a surrounding red box in our DIME models.

In our designs the layout of secure and critical SIBs is not different from the layout of the insecure SIBs within DIME. This is so as our aim is to make the task of switching between insecure and secure SIBs the easiest possible even for users which are not at ease with principles behind the XMDD. However, secure and critical SIBs are clearly contained in dedicated palettes, and so are easily distinguishable. As a rule of security, we impose that each SLG must be identified with an overall security level which cannot be lower than the one of any of its constituting SIBs. For instance, a SLG containing a second level SIB cannot be seen as a SIB of first level from any of the higher hierarchy level SLGs.

IV. Secure remote control of anthropomorphic actuators

The Human-Machine Interaction (HMI) paradigm is traditionally dominated by direct manipulation and physical interfaces (e.g., gloves, keyboard, joystick). These interfaces are often cumbersome and inadequate, and require dedicated training phases and calibration procedures. In addition, they

Fig. 1. Pipeline of the application: a sign (letter "V" from Italian Sign Language in the example) is performed by a signer, processed by the vision-based algorithms and set to be remotely reproduced by the robotic hand, so that a deaf-blind recipient can remotely interpret the carried message (the reader is referred to [18] for more details).

are often not suitable for people with disabilities. A valid alternative comes from modern motion tracking technologies, which instead offer many advantages in terms of usability, reduced costs and learning time, and do not require calibration procedures. In the past, we have already investigated vision-based interfaces for remote control of machine and robotic actuators in assistive [18] and rehabilitative [19] applications.

A. The Models

We present a case study based on a gesture-based communication pipeline to remotely control robotic actuators (e.g., an exoskeleton) leveraging security primitives and secure SIBs with the aim of guaranteeing confidentiality of the exchanged data and security, meaning that robotic actuators can receive and accept only valid inputs coming from authorized devices. The secure pipeline has been thought for applications in the field of remote communication among deaf-blind [18], but it could also be employed in similar domains or in other fields such as tele-rehabilitation.

An example of another possible usage of the secure communication pipeline is to enforce a remote communication system for deaf-blind people. Such "tele-signers" require signs from tactile Sign Languages to be recognized on one side resorting to vision applications and to be reproduced on the other side resorting to anthropomorphic actuators that are intelligible to the deaf-blind recipient. The pipeline is represented in Fig. 1. The reader may refer to [18] for further details.

For each operation which needs security, and especially for mission critical operations, two versions of the corresponding SIB have been designed: the "regular" one (i.e., without particular attention to security aspects) and the secure one. Each SLG in every pair is easily interchangeable in DIME by simple replacement via drag and drop; acting like this, it is possible to secure even complex SLGs and models in a few minutes by simply replacing critical sub-parts with their secure version. SLGs defining the models are organized in a hierarchical way, in order to improve their readability and especially their reusability. In fact, SLGs consist of an assembly of orchestrated SIBs as explained in [10], but can also be used themselves as higher level building blocks within higher level SLGs. This refinement by hierarchy concept is very powerful. It helps in particular to manage distributed development (when certain SIB palettes belong to a certain team or provenance) and distributed management and maintenance concerns, supported by rich and semantic interface definitions that are a form of contracts and Service Logic Agreements (SLAs) expressed by means of properties.

In our concrete application, the communication we wish to secure is based on the Robot Operating System (ROS). ROS [20] is an open-source, meta-operating system for robot software development that provides a collection of packages, software building tools, and an architecture for distributed inter-process and inter-machine communication. The building blocks of ROS-based applications are called nodes. A node is a piece of code which implements a specific functionality, described in a proper SLG within the XMDD framework. Nodes interact with each other by subscribing or publishing messages on specific ROS topics. The communication between nodes is based on the TCP network protocol, thus native ROS does not guarantee any security.

The lowest level in the hierarchy is represented by the ROS node which has access to the input device (Microsoft Kinect or Leap Motion) to extract data that will be processed later on. Since data are represented from joints identifying human hand skeleton, the node is named `SkeletonExtractorNode`. The DIME model of such a node is shown in Fig. 2. It is possible to notice the clarity of such a design in comparison with code. This makes the model understandable also for ROS experts with minimal, or none, expertise of the actual application. Each ROS node has to be initialized with a minimal set of parameters including a string identifying its name, a string identifying the ROS package name and a Boolean node indicating whether the node itself presents a unique name or it must be anonymized. Initializing a ROS

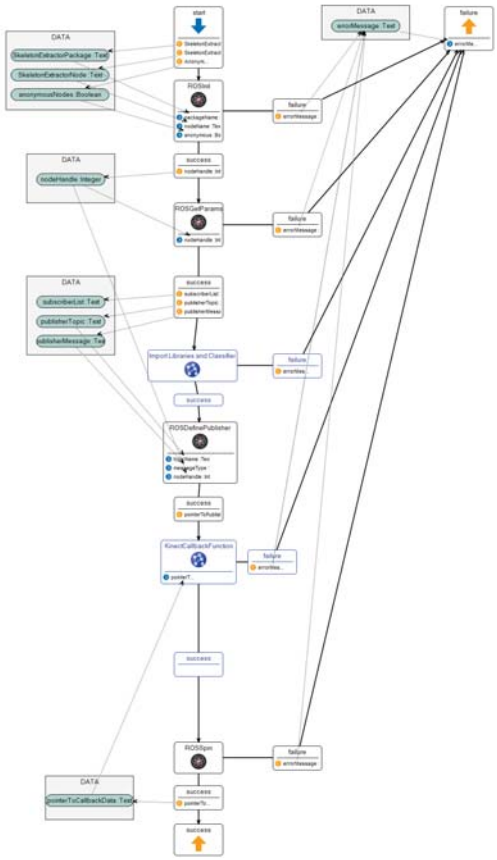


Fig. 2. Control and Data flow of the `SkeletonExtractorNode` for the Microsoft Kinect device.

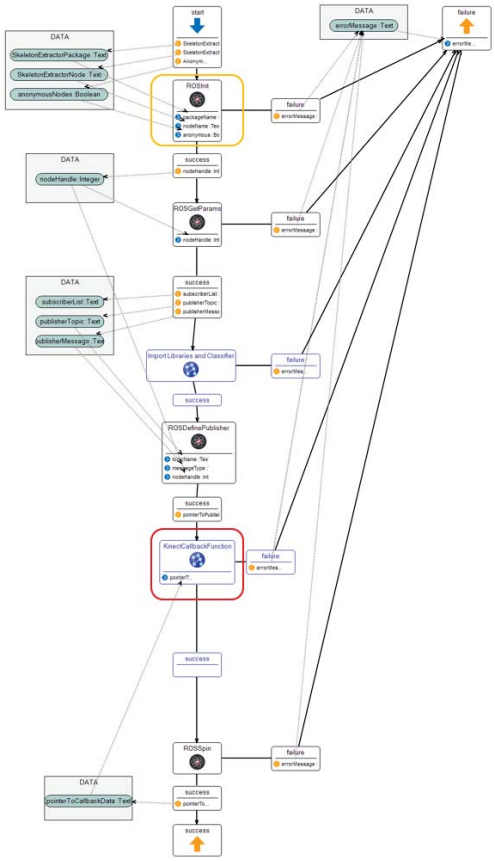


Fig. 3. Control and Data flow of the `SkeletonExtractorNode` for the Microsoft Kinect device with secure SIBs. In particular, please note the yellow- and red-boxed SIBs.

node is done by publishing its existence to the network of intra- and inter-machine communications, to let other nodes communicate with it via message exchanging on topics.

The first SIB right after the start SIB is named `ROSInit`, it is in charge of such an initialization, and represents also the first example of SIB for which we have provided a secure version.

This SIB is defined in DIME as follows:

```
sib ROSInit :
communication_pipeline.ROS#rosInit
     packageName : text
     nodeName : text
     anonymous : boolean
    -> success
     nodeHandle : integer
    -> failure
     errormessage : text
```

meaning that this SIB is defined in the class ROS within the Java package `communication_pipeline`.

This SIB accepts three input parameters ("packageName", "nodeName" and "anonymous") and provides two output branches: when errors or exceptions occur (e.g., it was not possible to communicate with the ROS network, maybe because ROS environment is not present in the machine executing the node) the "failure" branch is executed, returning "errorMessage" as output; otherwise, the node has been correctly instantiated and a handle will be returned as outcome of the success branch. Note that SLGs are orchestrations, meaning that (unless there is fork/join parallelism) there is a single threaded execution. For this reason, upon execution SIBs activate only one of their outgoing branches, namely the one corresponding to the right continuation in the control flow according to their execution's outcome.

In the Data model view, when failures occur, the corresponding error message is put in an appropriate variable (as shown in the top data container in Fig. 3) and the failure outgoing branch of the SLG is executed, causing in this case the error message to be printed and the control flow to return to the upper level in the hierarchy from the failure condition. We see here how easy it is to model the data flow in DIME, as data are connected and shared as any other resource. Note also

the independence of this representation from any particular paradigm of communication (equivalent to a PIM/CIM in traditional MDA): it can be a shared memory variable, a file store, or a messaging mechanism. The precise nature can be designed by refining it to a specific technological option.

Otherwise, the control flow proceeds by executing the subsequent SIB, named ROSGetParams, which identifies and defines proper parameters of ROS to ensure the correct execution of the SLG, i.e., of the node itself. The `ROSInit` node is not a mission critical node and has not been represented with a secure SIB in Fig. 2. Even if it failed due to an attack, the whole ROS communication pipeline would not start, thus the attacker could not spoil information from the input devices nor command maliciously the output robotic interfaces. Nevertheless, this node requires inter-machine communication, since data from the node itself (e.g., its name) and from the user launching it have to be propagated through the ROS network, which is not intrinsically secure. For this reason, we have prepared a secure version of this SIB. Even if the layout of the SIB does not change, so that the two `ROSInit` SIBs are easily interchangeable, their implementation is different: in fact, the secure SIB will rely on a trusted network for data exchange, and will check whether the ROS environment is consistent and coherent (by checking that its version number is correct) and whether the user is authorized to initialize ROS nodes on that particular ROS environment.

The authorization is defined in a fashion similar to the example and discussion in [10]. Similarly to the SIB organization and management explained in [13] on the cryptography example, the secure SIB is defined within a different package (i.e., we have defined a secure package to substitute the `communication_pipeline` one) and a dedicated palette, so that it is immediate for users to distinguish them, and to choose between insecure and secure versions of critical SIBs.

B. Dealing with Security Properties

In this section, we focus on properties that have been identified by the designers of the application. We first list the properties and then discuss techniques that can be used to enforce their correctness.

1) Property 1: any data variable within the data flow must have a unique data type, which has to be coherent and fixed through all the life of the variable itself;
2) Property 2: any SIB and any SLG which expect a set of input parameters must be executed with a set of parameters that fit the expected one for its size, and each of these parameter must be actually linked to a data variable of a proper type;
3) Property 3: any SIB and any SLG must provide at least one outgoing branch, and can terminate with the execution of only one outgoing branch;
4) Property 4: any SIB and any SLG which provide a set of output parameters must return a set of parameters that fit the expected one for its size, and each of these parameter must be actually linkable to a data variable of a proper type;
5) Property 5: it is not possible to initialize ROS nodes if a proper version of the ROS environment is not installed and active;
6) Property 6: it is not possible to execute ROS nodes that have not been initialized;
7) Property 7: it is not possible to subscribe to unadvertised ROS topics;
8) Property 8: it is not possible to read from ROS topics if publishers have not written data;
9) Property 9: it is not possible to read from input devices information in a format different from the one they support;
10) Property 10: it is not possible to send commands to output interfaces if no data has been read from input
11) Property 11: it is not possible to receive input data at a frame rate higher than the one supported from the input device; interfaces;
12) Property 12: it is not possible to send commands to output interfaces starting from data not acquired from ROS and the input devices;
13) Property 13: it is not possible to send to output interfaces configurations that they cannot reproduce;
14) Property 14: access to hardware interfaces must be realized through secure primitives;
15) Property 15: hardware interfaces must be checked against a list of valid supported devices.

C. On enforcing properties

Formal verification and software quality assurance offer a rich variety of techniques to enforce security/safety properties. Examples of such techniques include model checking, testing, runtime verification techniques, and synthesis or correct by construction techniques.

1) **Model Checking** [21] allows us to verify a model of the system and all its executions with respect to some properties expressed in a logic, frequently resorting to temporal logics. Those logics allow the expression of predicate-based assumptions on states combined with temporal ones on the sequence of states (e.g., eventually, the state `q` will satisfy the predicate). Those techniques are used at design time, and use the knowledge of the entire model that they are able to see. For example, software model checking checks properties of models of the code (this is useful e.g., to verify for example the correctness or security of SIB implementations), while behavioral or system-level model checking is more abstract, and sees behavioral elements (components, services, or in our case SIBs) as uninterpreted propositions, thus can check very efficiently their correct compositions. Model checking is successful and efficient for boolean properties (yes/no questions), posed to a large variety of system model types with a wide range of expressiveness, but has difficulties with other types of variables, albeit SMT-solvers [22] and systems with integrated decision procedures (e.g., for arithmetics) have helped to make progresses there. Examples of successful classical model checkers include

Spin [23] for LTL properties and SMV [24] for CTL, or Java Pathfinder [25] as a software model checker for Java. Being interested in a deep analysis and diagnostic capability at the level of SLGs as service orchestration models in which atomic SIBs are elementary entities, we do not suffer the state explosion problem normally experienced with fine grained models. With these system characteristics in mind, we will mostly use GEAR [26], which is an explicit state model checker for modal mu-calculus with game based diagnosis facilities. For probabilistic properties we envisage to use.

2) **Testing techniques** exercise the systems with respect to a series of inputs, or test vectors. The main challenge here is to make sure that those inputs will cover as much behaviors as possible [27]. The choice of those inputs is random when we have no knowledge on the design, but can become symbolic or concolic [28] when the system is executable (mostly meaning in software that the code must be available). Microsoft is one of the most demanding users for such techniques. They are also largely used in vulnerability and malware analysis. Depending on the phase when they are deployed (at design time, execution time, etc), those techniques may or may not assume the full knowledge on the environment. In terms of testing, there are categories of systems that can be addressed in a specialized fashion. Leveraging the techniques of [29], in our platform, for example, DyWA provides fully automated test facilities for web applications. This way we can combine unit test of the implementations at the SIB level with traditional model driven testing at the SLG level, and additionally test the user facing behavior with the DyWA facility in automated fashion.
3) **Runtime techniques** [30], [31] allow us to monitor an execution of the system and eventually take decisions in case some red light is crossed. One of the best example of the usefulness of such techniques is in anti-virus and malware detection, where the execution of the code is monitored until completion, or until a series of flags have been passed (in which case an exception is thrown). Those techniques are of interest when we only have a partial view of the system and we have no prior knowledge on the environment that is used. Their require that we can instrument the system so that the necessary information can be monitored. This can be done by modifying the code in a white or grey box fashion, in case we have the code and thus can augment it with runtime assertions or checks, for example in JML for Java programs, or if we are able to overlay the monitoring as a woven aspect in an aspect oriented development paradigm. If this is not possible, a black box approach includes having a model of the behaviors one wish to monitor run in parallel with the actual system under consideration, and use the (usually state-transition) model as an abstract indicator/predictor of the health of the system, stopping the execution when the model foresees some "safety" threshold trespassing. Of course, in this case one learns by failure, as those models can be only refined and improved after having experienced their inadequacy.
4) The **correct by construction** paradigm is attractive because instead of leading to repairing errors, be it in some system models at design time or in the system itself at runtime, it helps prevent the insurgence of non-conform behaviour at all. It assumes that there are strong hypotheses on the "knowledge" these approaches have of the system under design and its elements, either in terms of the system itself (code if software, the device if hardware), or of properties guaranteed by a trusted third party. For example, depending on what one has and what one trusts, termination properties of a SIB can be ensured by code inspection, software model checking, by testing, or by experience of the community (if it is a widely used library and has been debugged by myriad users). It also assumes that there is total control on the (quality of the) tools that are used to develop the system. These hypotheses are there to guarantee that the system will satisfy certain properties at design time because it has been built from certain conforming entities and using techniques that are known to enforce or preserve those properties of the product. As an example, the BIP toolset guarantees that if two components are secured, then this will also be the case of their composition, this being achieved by restricting the way components can interact together. Other examples are on restricting the high-level languages to those whose type can be controlled. For example, this typically excludes programming languages typed at runtime as unsuitable, but also prohibits constructs that may give rise to some risks, which is the motivation behind the existence of MISRA C [32]. Successful examples of correct-by-construction are high-level hardware synthesis [33], as well as workflow and process synthesis [34], [35]: they have in common that they do not strive to create the circuit or the code from scratch, but (like in our approach) they correctly choose, configure, assemble and wire their respective products starting from libraries of elements with characterized properties, that have been established independently.

With this in mind, the choice of which techniques to adopt and where (the scope) and when (in which phase of the product construction) to enforce them largely depends on various criteria that include (but are not limited to) 1. the nature of the property (behavioral, structural, involving variables...), 2. the best moment to verify the property (at design time, at deployment time...), 3. our knowledge of the environment (do we have an estimate of all the variables, do we have knowledge on the average user), and 4. the restrictions we permit on the design process (e.g., is this reasonable to forbid the user to write in python).

We now briefly examine our properties and discuss which technique could be applied.

We first observe that we have full control on Property 1 to Property 4: indeed, they are environment-independent, and

concern things (SLGs and SIBs) that live within DIME, which we control ourselves. We can thus enforce these properties by construction, acting on DIME. We distinguish two categories of properties: correct typing, and morphology.

1) **Properties 1, 2, and 4** concern **correct typing**. They are independent of the particular application and of the application domain: they should apply to every SIB and every SLG. As we describe in [11], through the use of CINCO and of DyWA, they are ensured by construction on any application built using the framework. This is an example of the advantage for designers of using frameworks with rich formal semantics.
2) **Property 3** is a well-formedness property on the **morphology** of SIBs and SLGs at the graph level (independently of any meaning of its elements). This is taken care of by the design in Cinco of the DIME framework, as described in [10] and [11]. Here, we assume that the code used in each SIB terminates, and we only focus on the flow of operation at the SLG level, inter-SIBs.

Properties 5 to 15 on the contrary are application domain specific, dealing with ROS. Depending on our knowledge and control of the ROS, we will have to use different approaches.

We first observe that all those properties can be encoded with temporal logic. Indeed, they refer to sequences of operations and predicates on given states. However there are differences in the nature of information they manipulate. As an example, Property 6 makes hypothesis on the sequence of operations, while Property 5 looks similar, but also make assumption on the ROS version, which is not captured in the model. Accordingly, we are more likely to be able to verify Property 6 with Model Checking, while we can only do this for Property 5 if we have this knowledge on the environment. In case we lack this knowledge, we can only enforce the property at runtime, or at construction time by imposing that we only communicate with ROS of some specific version.

With this distinction in mind, Properties 7, 8, and 10 are similar to 6 and Properties 9 and 11 are similar to 5. The similarity between 6, 7, 8 and 10 is obvious, and we propose to model check them with our GEAR tool. The similarity between 5, 9 and 11 is more intriguing. In fact, Property 9 is a domain specific instance and refinement of property 2 and 4: information formats can be described by means of types. This information would be present in a (possibly ontological) description of the devices, making it possible to determine the format compatibility in a static way. Property 11 is strictly dependent on the input device used within the application. It can be verified by deriving a limit from the technical specifications of the input device (e.g., we know that a camera like Microsoft Kinect cannot provide more than 30 frames per second) and then checking that no one of the SIBs managing input data works at an higher frequency than this limit.

The last four properties are concerned with communication primitives and interfaces. This means that we will mostly enforce them by construction. As an example, Property 12 is concerned with the way ROS uses to communicate among nodes and the control flow within our models. By construction, it should not possible in our models to admit data which is not coming from authorized input devices and ROS nodes; external data may derive from attackers or unauthorized nodes or processes. On the other hand, Property 13 derives from a security request by the hardware output interface(s), that could be damaged (or cause problems or incidents) when forced to reproduce configurations that lay beyond a safe working area, defined within the output driver. Property 14 is actually a security request specification, and needs to be dealt with by overlaying a security layer on top of those primitives dealing with hardware, as explained in [13]. Also, there we have a need to identify and then deal with Data at rest/Data in motion aspects, along the lines of [12]. An easy way to enable this is to identify the "Hardware" SIBs with a hardware label (an atomic proposition, for the use in LTL or CTL properties), so that we can use this and other similar information in possibly layered properties. Finally, Property 15 seems to request a preprocessing that does this check: it would be either an extension to the SLG that does this, if it is performed every time the SLG is executed, or a distinct process e.g. in case the application is configuring a laboratory with a number of devices, and then these devices stay fixed for a while until a new reconfiguration happens.

V. Conclusion

In this paper we have discussed an approach to overcome difficulties and gaps which are typically encountered when deailng with security-oriented model-driven approaches. As state-of-the-art MDS approaches address security only at a late stage of development, they are not suitable for the needs of modern companies and industry in general, requiting a fast turnaround at low incremental costs. Instead, we showed how an OTA-based XMDD approach can systematically help integrate security ab-initio in applications created within the SEcube platform. Since our approach is based on a set of reusable SIBs organized within dedicated palettes in DIME, it decouples the issue of guaranteeing that the SIBs are correct and secure from the issue of analyzing the applications, which can be greatly simplified by knowing the characterization of each SIB in advance. We applied our approach to the concrete realm of computer vision steering robotics, presented the safety and security properties elicited on the specific case study, and discussed the ways they can be enforced.

We think that this work paves the way to a novel user-centered programming paradigm, allowing an easy, and almost hidden, integration of security domain-dependent, but application-independent, properties in almost any field of modern IT, that could be of great interest for companies and industry.

Acknowledgment

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094, ICT COST Action IC1204: Trustworthy Manufacturing and Utilization of Secure Devices, and co-funded under the European Regional Development

Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

[1] Tiziana Margaria and Bernhard Steffen. Continuous Model-Driven Engineering. *IEEE Computer*, 42(10):106—109, October 2009.

[2] Krishnakumar Balasubramanian, Aniruddha Gokhale, Gabor Karsai, Janos Sztipanovits, and Sandeep Neema. Developing applications using model-driven design environments. *Computer*, 39(2):33–40, 2006.

[3] Phu H. Nguyen, Jacques Klein, and Yves Le Traon. Model-driven security with a system of aspect-oriented security design patterns. In *Proceedings of the 2Nd Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling*, VAO '14, pages 51:51–51:54, New York, NY, USA, 2014. ACM.

[4] Premkumar T Devanbu and Stuart Stubblebine. Software engineering for security: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 227–239. ACM, 2000.

[5] Tiziana Margaria and Bernhard Steffen. Service-orientation: conquering complexity with xmdd. In *Conquering Complexity*, pages 217–236. Springer, 2012.

[6] Tiziana Margaria and Bernhard Steffen. Service-Orientation: Conquering Complexity with XMDD. In Mike Hinchey and Lorcan Coyle, editors, *Conquering Complexity*, pages 217–236. Springer London, 2012.

[7] Tiziana Margaria and Bernhard Steffen. Agile IT: Thinking in User-Centric Models. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation*, volume 17 of *Communications in Computer and Information Science*, pages 490–502. Springer Berlin / Heidelberg, 2009.

[8] Bernhard Steffen, Tiziana Margaria, Ralf Nagel, Sven Jörges, and Christian Kubczak. Model-driven development with the jabc. In *Hardware and Software, Verification and Testing*, pages 92–108. Springer, 2006.

[9] Tiziana Margaria and Bernhard Steffen. Business Process Modelling in the jABC: The One-Thing-Approach. In *Handbook of Research on Business Process Modeling*. IGI Global, 2009.

[10] Steve Boßelmann, Johannes Neubauer, Stefan Naujokat, and Bernhard Steffen. Model driven design of secure high assurance systems: an introduction to the open platform from the user perspective. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[11] Stefan Naujokat, Michael Lybecait, Dawid Kopetzki, and Bernhard Steffen. Cinco: A simplicity-driven approach to full generation of domain-specific graphical modeling tools. *Int. Journal on Software Tools for Technology Transfer (STTT), Springer Verlag, (to appear)*, 2016.

[12] Antonio Varriale, Paolo Prinetto, Alberto Carelli, and Pascal Trotta. Secube^{TM}: Data at rest & data in motion protection. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[13] Giorgio Di Natale, Alberto Carelli, Pascal Trotta, and Tiziana Margaria. Model driven design of crypto primitives and processes. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[14] Tiziana Margaria, Bernhard Steffen, and Manfred Reitenspieß. Service-oriented design: The roots. In *Service-Oriented Computing-ICSOC 2005*, pages 450–464. Springer Verlag, 2005.

[15] Antonio Varriale, Giorgio Di Natale, Paolo Prinetto, Bernhard Steffen, and Tiziana Margaria. Secube^{TM}: An open security platform: General approach and strategies. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), July 2016, in press.

[16] David Basin, Jürgen Doser, and Torsten Lodderstedt. Model driven security: From uml models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 15(1):39–91, 2006.

[17] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[18] Ludovico Orlando Russo, Giuseppe Airò Farulla, Daniele Pianu, Alice Rita Salgarella, Marco Controzzi, Christian Cipriani, Calogero Maria Oddo, Carlo Geraci, Stefano Rosa, and Marco Indaco. Parloma-a novel human-robot interaction system for deaf-blind remote communication. *International Journal of Advanced Robotic Systems*, 12, 2015.

[19] Giuseppe Airò Farulla, Daniele Pianu, Marco Cempini, Mario Cortese, Ludovico O Russo, Marco Indaco, Roberto Nerino, Antonio Chimienti, Calogero M Oddo, and Nicola Vitiello. Vision-based pose estimation for robot-mediated hand telerehabilitation. *Sensors*, 16(2):208, 2016.

[20] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system in: Icra workshop on open source software. *IEEE, Kobe, Japan*, 2009.

[21] Christel Baier, Joost-Pieter Katoen, et al. *Principles of model checking*, volume 26202649. MIT press Cambridge, 2008.

[22] Clark W Barrett, Roberto Sebastiani, Sanjit A Seshia, and Cesare Tinelli. Satisfiability modulo theories. *Handbook of satisfiability*, 185:825–885, 2009.

[23] Gerard J Holzmann. The model checker spin. *IEEE Transactions on software engineering*, 23(5):279, 1997.

[24] Jerry R Burch, Edmund M Clarke, Kenneth L McMillan, David L Dill, and Lain-Jinn Hwang. Symbolic model checking: 1020 states and beyond. *Information and computation*, 98(2):142–170, 1992.

[25] Willem Visser, Corina S Psreanu, and Sarfraz Khurshid. Test input generation with java pathfinder. *ACM SIGSOFT Software Engineering Notes*, 29(4):97–107, 2004.

[26] Marco Bakera and Clemens Renner. GEAR: Game-based, Easy And Reverse model-checking. http://jabc.cs.tu-dortmund.de/modelchecking/, 2008. [Online; accessed 19-May-2016].

[27] Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner. *Model-based testing of reactive systems: advanced lectures*, volume 3472. Springer, 2005.

[28] Koushik Sen, Darko Marinov, and Gul Agha. *CUTE: a concolic unit testing engine for C*, volume 30. ACM, 2005.

[29] Stephan Windmüller, Johannes Neubauer, Bernhard Steffen, Falk Howar, and Oliver Bauer. Active continuous quality control. In *Proceedings of the 16th International ACM SIGSOFT Symposium on Component-based Software Engineering*, CBSE '13, pages 111–120, New York, NY, USA, 2013. ACM.

[30] Joost-Pieter Katoen and Perdita Stevens, editors. *Tools and Algorithms for the Construction and Analysis of Systems, 8th International Conference, TACAS 2002, Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2002, Grenoble, France, April 8-12, 2002, Proceedings*, volume 2280 of *Lecture Notes in Computer Science*. Springer, 2002.

[31] Klaus Havelund and Grigore Rosu. Synthesizing monitors for safety properties. In Katoen and Stevens [30], pages 342–356.

[32] MIRA Ltd. MISRA-C. http://www.misra-c.com/, 2013. [Online; accessed 19-May-2016].

[33] Michael C McFarland, Alice C Parker, and Raul Camposano. The high-level synthesis of digital systems. *Proceedings of the IEEE*, 78(2):301–318, 1990.

[34] Sven Jörges, Anna-Lena Lamprecht, Tiziana Margaria, Ina Schaefer, and Bernhard Steffen. A constraint-based variability modeling framework. *International Journal on Software Tools for Technology Transfer*, 14(5):511–530, 2012.

[35] Bernhard Steffen, Tiziana Margaria, and Burkhard Freitag. Module configuration by minimal model construction. 1993.

Italian National Cyber Security Framework

Roberto Baldoni
Cyber Intelligence and Information Security
Research Center
Sapienza University of Rome
Cyber Security National Laboratory
baldoni@dis.uniroma1.it

Luca Montanari
Cyber Intelligence and Information Security
Research Center
Sapienza University of Rome
Cyber Security National Laboratory
montanari@dis.uniroma1.it

***Abstract*—Despite informative systems are became the key in the management of both critical infrastructure assets like power grid, industrial systems, transportations, etc. and of every kind of business, from small to large, the cyberspace and its essential components are still exposed to a variety of risks. First, there is the steady presence of vulnerabilities: today it is not possible to have systems that are not vulnerable. The multitude of the 0-day attacks is a proof of this, the black markets proliferate, and thus it is mandatory to always take into account the cyber threats. The lack of awareness, at all the levels of organizations' chart, lets the cyber risk to be very relevant for the companies, at the same level of the financial or reputational risk.**

This paper presents the Italian National Cyber Security Framework, a methodology that aims to offer to the organizations a volunteer approach to cope with the cyber security risk, in order to manage it, reduce it, and be aware of it. The Framework approach is deeply tied to the risk analysis and not to the technical standards, it is a generalization of the US NIST Framework for Improving Critical Infrastructure Cybersecurity. It has been realized in alignment with the NIST and has been adopted by the Italian government.

I. Introduction

Nowadays the entire economy and the welfare services of a developed country are based on facilities and services provided through the cyber space, a cluster of interconnected and heterogeneous networks, protocols and IT applications all around us. IT accidents impacting such facilities and services may have huge economic consequences at national, enterprise and single citizen level. Such accidents impact not just the cybernetic framework, because they may start there and then reach the physical facilities too, causing even primary services to be unavailable, therefore leading to an economic loss or even human loss. Accidents may be normal or caused by terrorists, cybercriminals, activists and by foreign countries (cyber-warfare). In those cases, if the victim is a company, beside the damage to its image, it may suffer a huge financial damage: From a simple loss of competitiveness up to the complete loss of the strategic asset control (IPR, process methodologies, IT systems, etc.). In the case of a country, it may lead to a reduction of defensive capacity or even a loss of independency. For a citizen, the cyber threat may cause damage to rights and constitutional concerns such as life, physical integrity, fundamental freedom, including the right to confidentiality, beside other economic impacts. Cyber threats certainly cannot be faced by giving up the potentials offered by the IT systems and their interconnection within the network, thus loosing the increase of productivity and efficiency linked with computerization. The answer should be systematic, aimed at raising the citizens awareness, the "duty of care" of companies and the International "due diligence" of the country about the cyber threat. As reported in detail in an OECD document [8] and reiterated various times in our report in the last years [3], [4] , it is crucial that in this process of collective raising awareness, we shift from an idea of "IT system security" or "IT security" to that of "cyber threat management". This means, among other things, to define a process that respects the Constitution principles regarding, for example, the business activity management in order neither to contrast the social benefit nor to affect safety, freedom and human dignity. This consideration implies that the cyber security perspective is not to be seen just in technologic terms, but rather requires taking into account the overall legal and formal duties and the principles of social interest, into which the public and private framework need to converge. For this reason, the duty of protection should become part of the top management responsibility of an organization, as it requires a specific and accurate evaluation by the ones who have the direction and management power [9].

As any company risk, the cyber risk cannot be eliminated and therefore requires a series of coordinated actions to be taken in order manage it. Such actions involve the organization and technology departments of the company, in addition to the financial management of the risk, also through the establishment of a residual risk management strategy and a strategy to protect the company balance. Furthermore, the cyber risk is intrinsically highly dynamic. It changes as threats, technology and regulations change. To start approaching this issue in a way which is useful for the country system (State, enterprises and citizens) it is necessary to define a common ground, a Framework, in which the various production sectors, government agencies and regulated sectors can recognize their business, so to align their cyber security policies in a steadily developing process. To reach this aim *a common Framework should be first of all neutral both in terms of business risk management policies and in terms of technology*, so that each player could keep on using its own risk management tools, managing its technology assets while monitoring at the same time the compliance with sector standards.

ISBN: 1-60132-445-6, CSREA Press ©

This document presents a National Framework for cyber security aimed, firstly, at creating a common language to compare the business practices to prevent and tackle cyber risks. The Framework may help an enterprise to plan a cyber risk management strategy, developed over the time according to its business, size and other distinguishing and specific elements of the enterprise. The Framework adoption is voluntary.

The remainder of this paper will present the process that led from NIST Framework to the Italian Framework in Section II and the Italian National Cyber Security Framework in Section III. The main concepts added to the NIST Framework, namely the priority levels and maturity levels, are presented in Section IV. The advantages that a Nation could have by adopting a National Framework are presented in Section V while Section VII concludes the paper.

II. From NIST Framework to the Italian National Framework

The Framework we present is based on the "Framework for Improving Critical Infrastructure Cybersecurity" issued by the NIST [6], from which it inherits key concepts such as Framework Core, Framework Implementation Tier and Framework Profiles. Thus, it adopts the Function and Category system of the Framework Core, which in facts represents that common ground, the meeting point between Framework and company standards, both technical and risk management standards.

The choice to use the US Framework is based on the idea that the answer to cyber threats should provide an alignment at international level, not only at national level. This is also to allow the corporation to align their cyber security management processes in an easier way at international level.

The NIST Framework offers a highly flexible framework, which is mostly targeted at crucial facilities; we developed it according to the characteristics of the social and economical system of our country, reaching a cross-sector framework that can be contextualized in specific production sectors or in company types with specific characteristics. This allows the transfer of practices and knowledge from one sector to another in an easy and efficient way. In this sense, we have introduced three important concepts in the Italian National Framework:

- Priority levels. The priority levels define which is the priority associated to every single Subcategory of the Framework Core. It should be noted that every organization is free to adapt its own priority levels according to type of business, size and own risk profile.
- Maturity levels. The maturity levels define the various ways in which every single Subcategory of the Framework Core can be implemented. The selected maturity level is to be carefully evaluated by each single enterprise according to its business and size, as well as its risk profile. Typically, higher maturity levels require greater effort both in financial and management terms. For some Subcategories it is not possible to establish maturity levels.
- Framework contextualization. Creating a contextualization of the Framework (for a productive sector, for business type or a single business), means (i)to define the set of Subcategories that are relevant and (ii)to define the priority and maturity levels appropriate for the implementation context. It is important to remark that the priority and maturity levels are defined during the creation of one contextualization of the Framework.

Each organization can adjust its own cyber security policies to its own business, risk tolerance and available resources, by defining the residual risk management strategies. This concept is expressed by the notion of *current profile* of the organization. The current profile is created by comparing the existing cyber security practices with the Framework Subcategories and related maturity levels. Through this comparison, the Subcategories that are already implemented by the existing practices with related maturity level are selected. This selection creates the current profile, to be compared with the *target profile*. The target profile consists in the selection of Subcategories and of the desired maturity levels, according to the organization needs. To have a current and target profile favors the gap analysis process and the definition of a roadmap to be followed in order to obtain the target cyber security level. In establishing the roadmap, the Subcategories with *high priorities* are the first to be implemented. Subcategories with *medium priority* and *low priority* have to be selected according to ones own needs and then implemented.

Figure 1 shows the relationship between the National Cyber Security Framework and the specific characteristics of an organization: implemented Enterprise Risk Management practices, applied IT security standards with related certifications, organization size and production sectors. In particular, the Framework, on a higher level of abstraction, serves as a bridge between the Enterprise Risk Management tools and the IT & Security Standards. The Figure shows the productive sector contextualization and the contextualizations based on the type of company. It should be noted that for each productive sector and each company type, various contextualization can be defined. We point out that the National Cyber Security Framework is not a static document, but rather a live one, which has to be updated according to the evolution of the threat, of technologies, of cyber security and of the risk management techniques. Such update should be censure by institutional competent bodies for its maintenance over the time.

III. The Framework

The NIST Framework core is made up of 21 Categories and 98 Subcategories, structured into 5 Functions. Each Subcategory represents a recommendation area, that the organization may decide to implement, if necessary by referring to the specific sector standard or regulation. The NIST Framework provides references to existing standards and Frameworks for each Subcategory: It is a partial mapping that covers most of the references already implemented by International organizations,

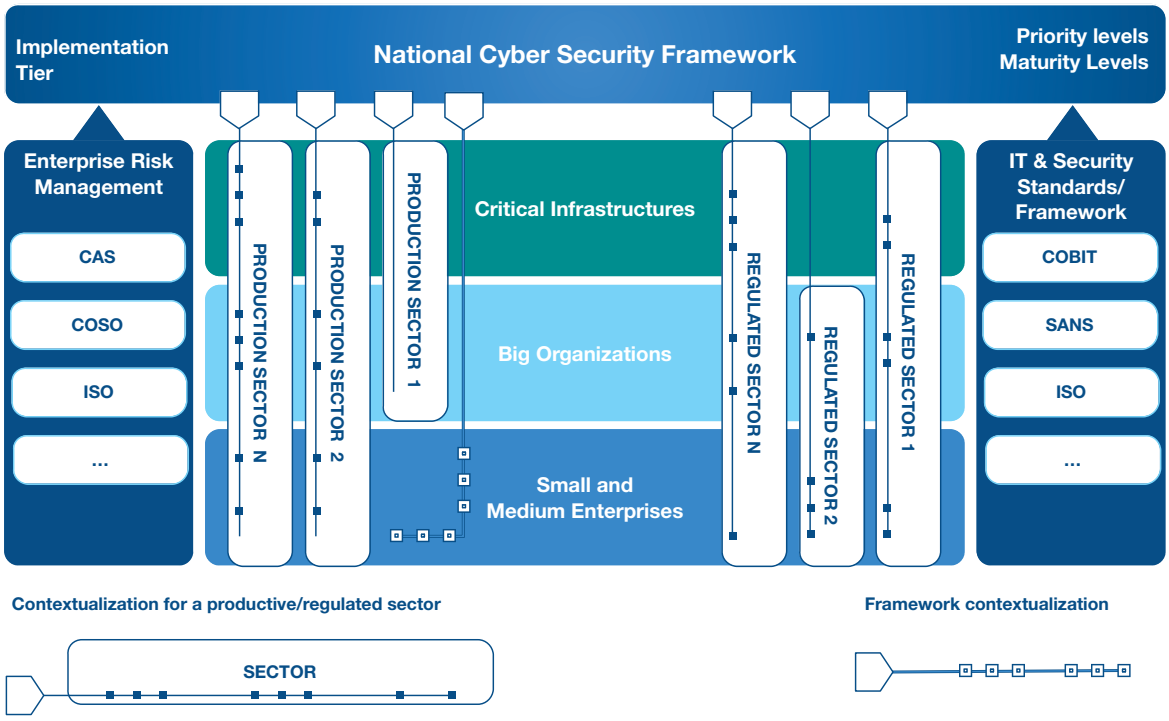


Fig. 1. National Cyber Security Framework and its link to the enterprise risk management, IT security standard, enterprise size and production sectors

such as the NIST Standard, the ISO/IEC and the COBIT Standards.

The National Framework extends such structure by introducing two new concepts: priority levels and maturity levels, defined during the creation of a contextualization. These two concepts allow to take account of the economic structure of our country, which is made of dozens of big companies and Critical Infrastructures and many small enterprises, therefore the Framework is suitable for SMEs, but remains targeted to Large Enterprises and Critical Infrastructures.

A. Framework Core, Profile and Implementation Tier

The Italian National Framework derives three fundamental concepts from the NIST Framework: Framework Core, Profile and Implementation Tier. Below they are briefly described, for further details, refer to the original document [6].

Framework Core. The core represents the life cycle structure of the management process of cyber security, both from a technical and organizational point of view. The core is structured hierarchically into Function, Category and Subcategory. Concurrent and continuous Functions are: Identify, Protect, Detect, Respond, Recover and they represent the main topics to deal with in order to strategically obtain an appropriate cyber risk management. Thus, the Framework, for each Function, Category and Subcategory, which provide information in terms of specific resources, defines the processes and technologies to be put in place in order to manage the single Function. Finally, the Framework core structure shows *informative reference*, informative references that link the single Subcategory to a number of known security practices by using sector standards (ISO, sp800-53r4, COBIT-5, SANS20 and others). The Framework Core structure of the NIST is showed by Figure 2.

Functions	Categories	Subcategories	Informative References
IDENTIFY			
PROTECT			
DETECT			
RESPOND			
RECOVER			

Fig. 2. Framework Core structure (from [6])

Profile. Profiles represent the result of the selection made by

an organization, of specific Subcategories of the Framework. Such selection can be performed according to various factors, that are mainly linked to risk assessment, business context and applicability of the various Subcategories. Profiles can be used as an opportunity to improve the security status by comparing an actual profile (also called current profile), with the wished profile (also called target). In order to develop a profile, an organization has to analyse each of the Subcategories and, according to the business driver and evaluation of ones own risks, to establish which ones have to be implemented and which ones are applicable to ones own context. Subcategories can be integrated with further practices, that are not provided by the Framework, for a complete risk management. The actual profile can therefore be used to define priorities and to measure the advancement towards the target profile. Profiles can be used also to perform a self-evaluation or to communicate ones own risk management level within or outside the organization. Finally, it should not be underestimated its use to define minimum profiles required by an organization in order to benefit from services provided by third parties. This use strengthens the entire supply chain in case of specific critical issues.

Implementation Tier. The implementation Tiers provide a context on how the company, as a whole, considers cyber risk and processes to manage it. There are four evaluation levels, from the softest to the hardest one: (1) Partial, (2) Informed, (3) Repeatable, (4) Adaptive. In particular:

Partial. The cyber security risk management of an organization is partial if it does not systematically take account of cyber risk and environmental threats.

Informed. The cyber risk management practices of an organization are informed if the organization has internal processes that take account of the cyber risk, but they do not cover the entire organization.

Repeatable. The cyber risk management model of an organization is repeatable if the organization regularly updates its own cyber security practices based on the risk management process output.

Adaptive. The cyber risk management model of an organization is adaptive if the organization frequently adjusts its cyber security practices by using its past experiences and risk indicators.

IV. Priority levels and Maturity levels

As stated before, we have introduced priority levels and maturity levels to the original NIST Framework's structure presented in Figure 2. These two concept are defined for each contextualization. In this section a description of priority levels, maturity levels and of the contextualization methodology is provided.

A. Priority levels

The priority levels help to support organizations and companies in the preliminary identification of Subcategories to be implemented in order to further reduce their risk levels, while balancing the effort to implement them. The Framework suggests the use of a priority scale of three levels among Subcategories. The objective is to:

- Simplify the identification of essential Subcategories to be immediately and binding implemented;
- Support the organizations in their risk analysis and management process.

The identification of priority levels assigned to Subcategories should be performed according to two specific criteria:

- Ability to reduce cyber risk, by working on one or more key factors for the identification, that is:
  - Exposure to threats, intended as the set of factors that increase or diminish the threat probability;
  - Occurrence Probability, that is the frequency of the possible event of a threat over the time;
  - Impact on Business Operations and Company Assets, intended as the amount of damage resulting from the threat occurrence;
- Ease of Subcategory implementation, considering the technical and organizational maturity usually required to put in place specific countermeasures.

The combination of these two criteria allows the definition of three different priority levels:

- High Priority: Actions that enable the slight reduction of one of the three key factors of cyber risk. Such actions are prioritized and must be implemented irrespective of their implementation complexity;
- Medium Priority: Actions that enable the reduction of one of the three key factors of cyber risk, that are generally easily implementable.
- Low Priority: Actions that make possible to reduce one of the three key factors of the cyber risk and that are generally considered as hard to be implemented (e.g. significant organizational and/or infrastructural changes).

Note that some Subcategories assume a specific priority for the used contextualization or assume a specific priority according to the organization context (possibly based on the associated risk evaluation), therefore, each organization, by implementing the Framework or during the contextualization activity, may redefine the specific priority levels for each Subcategory.

B. Maturity levels

Maturity levels enable the measurement of maturity of a security process, maturity of a specific technology implementation or an assessment of the amount of resources needed to implement a specific Subcategory.

Maturity levels provide a reference according to which each organization may evaluate its own Subcategory implementation and establish targets and priorities for their improvement. The levels must be incremental, from the lowest to the highest. Each level has to provide incremental practices and controls respect to the lower maturity level. Each organization will

evaluate the satisfaction of control in order to identify the maturity level that has been reached. For some Subcategories it could not be possible to define maturity levels. Within the definition of maturity levels, the following characteristics have to be taken into account:

- Specificity for Subcategory. An organization may have various maturity levels for different Subcategories;
- Completeness of security practices. The maturity level of a Subcategory corresponds at least to the one in which the related security practices are performed.

This enables to:

- Define partially or entirely ones own maturity level;
- Identify the target level: Partial or overall;
- Identify the necessary security practices in order to reach the target level.

The complete National Framework Core is depicted in Figure 3.

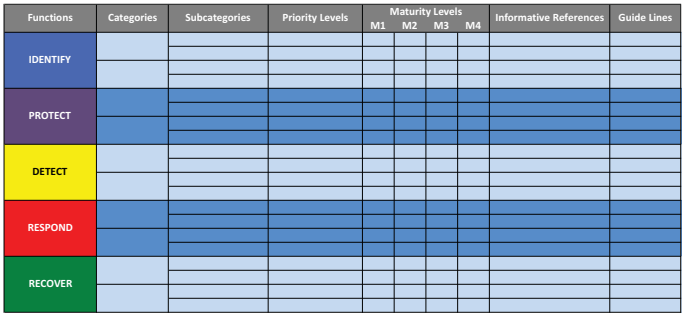


Fig. 3. National Framework with introduction of maturity levels and priority levels.

In general, the Framework provides just the rules to define the maturity and priority levels, as these and their related controls are extremely linked to the company nature, the business sector, the its structure and size, as well as to the business model. In terms of SMEs context, this document presents a specific contextualization, the priorities for this company segment and the minimum maturity level to be provided in order to raise ones own ability to manage the Cyber risk.

C. Contextualizations

Framework contextualization for a production sector or an homogenous category of organizations means to specify its core (i.e. to select the Functions, Categories and Subcategories) and to specify the priority and maturity levels for the selected Subcategories. Up to now, all notions have been introduced regardless, for example, of the production sector, type of employees, size and position of the organization on the territory. When a Framework is contextualized, all or some of the previously described elements are taken into account. A Framework contextualization is performed following the steps below:

1) select the list of relevant Functions/Categories/Subcategories for the organization according to all or some of the previous elements (production sector, size and territorial position of the organization, etc.);
2) define the priority levels for the implementation of the selected Subcategories;
3) define the guidelines at least for high priority Subcategories;
4) specify the maturity levels at least for high priority Subcategories;

All organizations that implement a specific Framework contextualization, must always implement high priority Subcategories, at least at a minimum maturity level.

The above steps must be implemented according to the specific business characteristics of the organization. Below is a list of the ones who can carry on the task of Framework contextualization. The Framework can be contextualized:

1) by the single company for the management of its cyber security program. This implies that the company is enough mature to manage the above steps and the following associated risk management model. For example, Intel was one of the first to provide a case study on how to contextualize the NIST National Framework for cyber security [5].
2) by an association of a production sector, in order to make the Framework contextualization available to all companies of the sector. This contextualization can also take into account the company size. For example, the IV work group of the CSRIC (The Communications Security, Reliability and Interoperability Council) provided a Framework contextualization for the communication sector, including producers of satellites, TV networks, landline networks and wireless networks in the United States [7].
3) by a sector regulator in order to make the Framework contextualization available to all organizations of the sector. Contextualization can also take into account the company size, beside the specificity of the regulated sector.
4) by any player that defines a Framework contextualization according to one ore more characteristics that the companies have in common, as for example geographic location, size, staff type, etc. A typical case may be a local group of small and medium enterprises that use services provided by a consortium. The latter can contextualize the Framework for that companies. Finally, this document describes in Part II a contextualization of the Framework for SMEs made by a mixed group of academics and IT security professionals. This contextualization is therefore part of this category.

It should be noted that every single organization, even if a contextualization is provided by a regulator or a sector organization, may define and include further Subcategories or specialize the existing ones according to its own business and cyber security targets

V. The advantages for the Italian context: SMEs, Large Enterprises and sector regulators

a) Small and Medium Enterprises.: The Italian framework is mostly made of SMEs, most of them have never faced the issue of IT security. This is generally due to a lack of cyber risk assessment: sometimes Small enterprises do not consider that they have information assets to protect, sometimes they do not know the many tools that modern hackers may use. The main issue for small enterprises, once they approach the security dimension, is represented by costs: They are not independently able to identify "quick-win" practices, which allow higher protection levels with minimum effort. As a consequence, these companies risk to make a wrong estimate of costs needed for their asset security, and this often make them give up the idea of improving security, with enormous consequent risks, which they are not aware of. The Framework provides a series of security practices that, especially for SMEs, are basic and economic at the same time. Such practices are called "high priority practices" and correspond to that set of operations, which bring the level of awareness, protection and therefore security to a basic value, which is sufficient for most of the Italian SMEs.

b) Large Enterprises.: The National Framework does not pretend to guide Large Enterprises or to replace their complex risk management. It can be yet very useful to support, through a unique method, the company risk management programs and processes, so to make them evolve consistently and in a structured way. Furthermore, Large Enterprises can benefit from the Framework for two fundamental aspects: its international nature and the possibility to require security profiles to their contractors. The Framework, indeed, is based on the NIST Framework, therefore it is fully compatible with the security profiles and assumes the international nature of the latter. As a consequence, it can favour the communication of its own security levels and known standards (as for example the ISO standards), but in an extremely cheaper way. From the contractors point of view, Large Enterprises and Critical Infrastructures may use the Framework to require given security levels to all or some of the players that form part of their supply chain, or just to the ones who have to deal with given resources. This mechanism increases the security of the entire enterprise environment and, as a consequence, minimizes the vulnerability to attacks.

c) Sector Regulators.: As far as sector regulators concerns, the National Framework provides ground for a clear and unique exchange, where it is possible to work consistently with regulatory companies as well as other regulators. The Framework may be used as a tool to define regulations and standards in a structured and compatible way together with other regulators. It enables the assessment of possible specific national, European and international regulations, general or specific ones, avoiding additional burdens and promoting the dialogue between regulator and regulated entities. Sector regulations, like all other regulations, remain effective, after their issue, for extremely long periods, compared to the evolution speed of the cyber threat. Therefore, it is important to establish review processes, especially for sectors, in which the security management is particularly crucial (e.g. bank sector, government agencies, etc.). The Framework can be used for a preliminary review of regulations at first, and then it is possible to follow the evolution of the Framework in order to update regulations and practices. Furthermore, establishing a mapping among sector rules and practices of the Framework represents a very useful exercise in order to point out the possible defects, which inevitably widen the attack territory for companies of ones own sector.

VI. Advantages for the country system: Towards an international due diligence

Considering that the economical, technologic and, consequently, the political aspects of cyberspace activities at international level will shortly become a crucial point of geopolitics, a national framework for cyber security is one of the elements that a country, as well as private companies under its jurisdiction, needs in order to secure the networks and IT systems. Beside the National Framework, further essential elements of a national system to increase the resilience to attacks are:

- an efficient CERT network: In 2014, Italy has created its own national CERT[1]. The national CERT supports citizens and enterprises through awareness raising, prevention and coordination of reaction to large scale cyber accidents. Furthermore, through a link with the other Government CERTs (CERT-PA for Public Administration and CERT-Defence), it can provide an updated overview on relevant events, that are useful to update and develop the enterprises cyber security programs;
- a system to share public-private (with bidirectional exchange) following the US ISAC pattern, in which enterprises of the same production sector or with very similar cyber risk exposure gather around joint working tables [2]. These round tables aim at preventing the cyber threat through appropriate intelligence actions;
- an integrated system of interactions between public-private-national research made of technology programmes, joint research centres, etc. [1], in which to find a technology reference point for defense and crisis management operations.

The single organization, besides interfacing with the previous elements, should implement internally the technology best practices that are typical of the IT risk management, such as: disaster recovery systems and business continuity of systems and networks, audit, vulnerability research of systems and security certifications of its own systems.

[1]The National CERT is available at http://certnazionale.it. The National CERT serves as aggregator and "certifier" of contributions, notification of highly reliable information coming from public and private, national and international entities. Enterprises can share, in a protected and safeguarded manner, all their information with the national CERT and with other validated subjects.

This framework of measures that go seamlessly from public to private, besides protecting our national economic interests, may be of crucial relevance within legal disputes between enterprises or international disputes among States, due to cyber attacks. Indeed, mitigating or worsening ones own position will depend on the "duty-of-care" or "negligence" that a State, company or both of them have followed over the time to minimize the cyber risk. From this point of view, the National Cyber Security Framework represents a tool to identify possible deficits within the cyber security management of an organization, both in the public and in the private sector and to define a risk management strategy that persists as the threat and technology change.

VII. Conclusion

In this paper we presented the Italian National Cyber Security Framework. The work that led to this document is the result of a public-private partnership composed by more than 30 persons, all of them united by the interest on the national cyber security awareness. The importance to have a common language that the National Framework defines, is fundamental to start a extremely formative process, able to enhance the cyber risk awareness. This awareness will contribute to achieve an enhancement of the national cyber security. It is clear that the common language can be used at several levels of complexity, from the micro-enterprises, at a minimum level, toward the critical infrastructure, where the Framework declinations will assume the maximum completeness level. Nonetheless, having a National Framework that put together different realities led the Italy to the reach of a good point of awareness. This awareness will be the key to get the cyber security problem of the organizations evolved, a problem that for too long has been ignored, especially in the perspective of the forthcoming fusion between the economy of the country and the cyber space. The reader is invited to download the National Framework document from www.cybersecurityframework.it.

VIII. Acknowledgment

The authors would like to thank all the people and organizations that collaborate in the hard task of the definition a National Framework. In particular, the other members of CIS-Sapienza and of the Italian National Cyber Security Laboratory, the governmental members of Presidency of Ministry councils, of Italian National CERT, of AgID, of Privacy Agency and of Italian Ministry for economic development, and the members of the panel of companies: AON, Deloitte, ENEL, ENI, Hermes Bay, KPMG, Intellium, PWC and Microsoft.

References

[1] Roberto Baldoni, Rocco De Nicola Editors, Il Futuro della Cyber Security in Italia, Consorzio Interuniversitario Nazionale Informatica, November 2015 https://www.consorzio-cini.it/labcs-home/libro-bianco

[2] Roberto Baldoni, Luisa Franchina, Luca Montanari. Verso una struttura nazionale di condivisione ed analisi delle informazioni. Franco Angeli Editore, 20 pages, ISBN 9788891706881, 2014.

[3] Roberto Baldoni, Luca Montanari Editors. 2013 Italian Cyber Security Report - Critical Infrastructure and Other Sensitive Sectors Readiness. Università degli Studi di Roma La Sapienza. 2014 ISBN 978-88-98533-13-8 http://www.dis.uniroma1.it/~cis/media/CIS\%20Resources/2013CIS-Report.pdf

[4] Roberto Baldoni, Luca Montanari Editors. 2014 Italian Cyber Security Report - Awareness, Defense and Organization in the Public Sector. Università degli Studi di Roma La Sapienza. November 2015 http://www.cis.uniroma1.it/csr2014

[5] Tim Casey, Kevin Fiftal, Kent Landfield, John Miller, Dennis Morgan, Brian Willis. The Cybersecurity Framework in Action: An Intel Use Case. Intel 2014 http://www.intel.com/content/www/us/en/government/cybersecurity-framework-in-action-use-case-brief.html

[6] Framework for Improving Critical Infrastructure Cybersecurity (Version 1.0) National Institute of Standards and Technology. 2014 http://www.nist.gov/cyberframework/

[7] Robert Mayer, Brian Allen (editors). Cybersecurity Risk Management and best practices:Final Report The Communications Security, Reliability and Interoperability (CSRIC) Council - Working Group 4, 2015 https://transition.fcc.gov/pshs/advisory/csric4/CSRIC_IV_WG4_Final_Report_031815.pdf

[8] Digital Security Risk Management for Economic and Social Prosperity: OECD Recommendation and Companion Document, OECD Publishing, Paris, 2015

[9] David Patt. Cyber security is not just the IT departments problem. Financial Times. November 2015. http://www.ft.com/intl/cms/s/0/f6b50038-92a1-11e5-bd82-c1fb87bef7af.html?desktop=true#axzz3srZntwJX

SESSION

SECURITY EDUCATION + INFORMATION ASSURANCE + HARDWARE SECURITY

Chair(s)

Prof. Ken Ferens

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

Towards a Mathematical Model for Autonomously Organizing Security Metric Ontologies

Gregory Vert
College of Security and Intelligence
Embry Riddle Aeronautical University
vertg@erau.edu

Bryce Barrette
Global Security and Intelligence Studies
Embry Riddle Aeronautical University
Barretb4@erau.edu

Bilal Gonen
Assistant Professor of Computer Science
University of West Florida
bilalgonen@gmail.com

Abstract:

Security metrics can be contextually related to the operation of the system and provide a static measure of system security. Some are better at determining the system security. However, their values can change dynamically to threat. Thus they can be grouped into ontologies measuring system health and security under threat conditions. The problem with current approaches is that they are static and that it does not consider the fact that metrics can also occur in multiple ontological classes. In this paper we develop and propose a model and mathematical grounds for ontologies that can dynamically reorganize based on the mathematical properties of the model. We initially develop a set of metrics that can then be incorporated into the ontological model that can autonomously reorganize into super and superior ontologies when threat is present. We identify sample of metrics to demonstrate the model on and evaluate the relations among metrics to develop the model. This work presents the initial relational model for autonomous recombination after developing the initial set of metrics and mapping them to ontological classes. This work is initial and conceptual with further work to be done in the future to empirically validate the model.

Keywords: Dynamic Organizing Ontologies, Security Metrics

I. INTRODUCTION

Computers are utilized in all major industries. Their value to these industries is incalculable and incredibly necessary to the function and continued profit of every industry that uses them. However, in return, cyber intrusion actively target these very computers and seek to defraud them of information or worse, disable their functions. Continued security improvements and infrastructure changes are necessary to maintain safety against these attacks. However, with numerous black markets for purchasing security exploit tools and hacking software's readily available, it is a constant effort to stay ahead of cyber-attack developments. These attacks continue to grow in strength and become more encompassing of network security [1]. This proves especially true in areas such as cloud computing. As modern industry can choose to move their infrastructures to offsite servers, there exist very few frameworks for security. As this is the case, very different security frameworks and standards need to be developed than what is standard today [2].

The methods for improving this security must be taken with several factors in mind. Increasing security without proper attention to other factors involved with the network may create network vulnerabilities. To obtain proper and complete security, a balance must be maintained between security improvements and functionality of the network system. This analysis must be carried over to cover multiple information systems and the configurations they may take. This is critical for improving systems that are vastly interconnected [3]. For industry networks that carry very large integrated systems, this balance is very necessary to maintain. Even if a system can operate very efficiently and conduct operations quickly, without proper security protecting it, a cyberattack event may bring the company down. This mutual exclusivity of both factors leading to a successful network emphasizes the importance of this balance. Protecting functionality while maintaining security is paramount.

The issue is the limitations of current security metric coverage and scope limit their contribution to current security needs. Security metrics are singular in their focus, do not interact and are limited in their communication to security personnel. Additionally, no known framework exists for combining different metrics together or determine relationships from these individual metrics. This causes the current metrics that exist to provide incomplete and uncomprehensive security monitoring and solutions. The framework of this paper purposes solutions to expand their coverage and usefulness when addressing cyber security.

II. BACKGROUND

One way a security manager can interpret events about a system is through metrics. Security metrics are used by today's professionals for determining the state of system security. Simply, metrics are a standard of measurement used to measure security factors within a network. [4]. As such, these metrics offer indicators for security management. By analyzing metrics, one can

find holes in the current systems, areas to improve, etc. this enables the incorporation of environmental factors into the creation of security metrics [5].

Research groups have attempted to develop security metrics over the years as they are an essential tool for system security maintenance and performance. However, the issue with current security metric systems is the static nature of these metrics. These metrics need to calculated and only accounts for a single indicator. While these are helpful for analyzing events, they are less helpful than they could be. Metrics should be taken and regrouped based on applicability to threat events in a system, creating much more descriptive and helpful metrics [6]. Ideally grouping would be autonomous and adaptive.

To enhance these metrics, adaptive methodologies can be applied to relevant security methods to improve their effectiveness. By adding adaptive methods, a much more tailored metric is produced for the security manager to interpret for their network security. Additionally, parameters can be fed into these security metrics to create a much more effective metric. The goal of adaptive metrics is to offer more effective indicators of system security that what is typical today. Such metrics should be developed and designed around multiple metrics and be based on a model that can be dynamically organized.

In addition, one issue that necessitates the use of adaptive security metrics is evaluating qualitative security methods to a quantitative measurable scale. This quantification is necessary to provide better understanding of complex phenomena and to enable informed decision making [7]. This enables security managers to be able to compare previous events in a uniform manner and easily compare with numerical readouts for their securities. In return, the manager can easily determine whether there is a need to decrease or increase security measures and in which areas to do so. Metrics of this quality can be used to assess network strength, find critical infrastructures of the network and suggest further coverage of future vulnerability based on past available historical data [8].

While there are defined metric systems, the goal of adaptive metrics is to create system that autonomously tailors itself to an individual system. [9]. To achieve this, there is a need to create metrics that are adaptable. To achieve this the following needs to be employed:

- Use policy to define security goals and properties
- Determine which objects within a security framework need to be protected
- Create methodologies that will collect and interpret data
- Develop a model for adaptive reorganization of metrics into groups of ontologies that can be utilized to measure system security under threat
- Develop framework for taking actions based on the results of the data.

The goal of this research is to define metrics that can be fit into ontologies of metrics that have mathematical properties and allow them to adaptively reorganize based on the types of threat present in a system.

The field is lacking autonomously defined models for security analysis [10]. These metrics will be a framework for implementation into the security framework. Additionally, the ontological model developed by this research enables easy applications to other areas of security in industry. In our initial research, we are developing the mathematics and the ontologies for creating relevant adaptive security metrics. This is referred as an adaptive security metric method (ASOMO).

The particular mathematics used for this work will be fuzzy sets. Fuzzy set theory in the developing model will allow metrics to be related to each other and thus placed in sets based on similarity. The fuzzy sets allow typical quantitative measures regarding security to be quantified in a meaningful way [12]. As an "underperforming" antivirus is a vague designation to compute, this makes analysis more difficult. However, when "underperforming" becomes a ".35" rating, mathematics can now be performed on metrics that otherwise could not.

Finally, the developing model will be an architecture or theory in practice. As the model's theory is developing based on this initial work, the metric data will be utilized to test if the model works mathematically. The rest of the paper presents an initial development of security metrics ASOMO model that can then be mapped into autonomously self-organizing ontologies of security metrics. The mathematical basis for reorganization is defined. This is initial and ongoing work. The next sections will detail an initial determination and classification of security metrics, organization into ontological classes, and evaluation of the relations among metrics. Finally it will present the initial mathematical model for self-organizing ontologies as threat sweeps through a system.

III. APPROACH

A. Class designations and ontological mapping

The goal of ontological class designations is to designate a framework in which metrics could exist for a particular threat. These metrics can apply to a particular scenario in which a security manager will select a class of metrics to meet his security needs and match a security environment.

In order to relate the metrics to their particular ontological classes, we are examining how to utilize fuzzy set theory to determine similarity and membership. The derived values are approximated values assigned to particular metrics and score on a system relating to their presence within a security state set. This value could be designated as a 1, pertaining an complete membership in in ontological set class. Otherwise, a 0 is a designation of having no relation to a set. By relating the highest scoring metrics within a particular system, an ontological grouping of security metrics can be created using this method.

The first phase of the model was to determine that classes of security metrics that would then be mapped onto the ontological model. Our initial ontological classification of metrics is the following:

(i) User End- This Ontology requires direct input from the user or workstation within a network. These metrics are usually intended from security manager influence.

(ii) Server End- These metrics are directly influenced by company security policy. These metrics can also exist entirely internally as well, affecting nothing else but the company network.

(iii) Physical- These metrics relate to physical, natural places or objects. This can also apply to real-life incorporeal concepts like temporal events or security atmosphere after terrorist attacks.

(iv)- Company Patch Risk- This metric covers the danger of a company-wide patch release. This is the amount of vulnerabilities to an entire network when a new patch is released.

(v)- Physical Security- These are the actual physical assets a company may own to offer protection to their employees and/or networking equipment.

(vi) Reliability- The expected time duration the system is operating before it fails in delivering its service.

(vii) Criticality- The importance of particular computers on a network. Derived from location of the computer, service and applications running, role of the computer and asset value of the computer.

(viii) Temporal- Events pertaining to time based events in the present, past and future. Events more current may be more relevant or vice versa

In order to determine degrees of similarity between the above ontological classes, we developed for the model fuzzy linguistic variables as shown in figure 1. These are initial proposed values equally dividing the range of 0,1 to make a statement of the models evaluation of degrees of similarities among ontological metric classes.

$$f(similiarity)\begin{Bmatrix} 1 = most\ similar\ (ms) \\ 0 = not\ similar\ (ns) \\ .5 = somewhat\ simliar\ (sws) \\ .25 = very\ unsimilar\ (vu) \\ .75 = \ very\ simliar\ (vs) \end{Bmatrix}$$

Figure 1. Fuzzy Set Linguistics

For example, the metric Company Patch Risk resembles a 1, resembling a completely similar to the ontology "Physical". This metric would then be assigned to the applicable ontology.

	Company Patch Risk	Physical security	Reliability	Criticality	Temporal
Sever End	vu	vu	sws	vs	vu
User End	vs	vu	sws	vs	vs
Physical	ms	vu	sws	vs	ns

TABLE 1. Ontological grouping by similarity

Table 2 represents the ontology similarity matrix. This will be used to determine super ontologies. This table takes two ontologies and, by using fuzzy set linguistics, their degree of relationship can be determined. The higher degree of relationship value found in the read outs of the ontology. The higher the degree of relationship, the stronger the two ontologies can be related. A high enough degree of relationship will designate them a super ontology and will be able to apply both to a security system equally.

	User End	Server End	Physical
Server End	ns	ms	sws
Physical	sws	sws	ms
User End	ms	ms	vs

Table 2. Similarity Matrix

By using the linguistic variables, an autonomous update process can be put place to dynamically calculate similarity. These groupings can also determine logical relationships between two metrics within an ontology.

Two high scoring metrics can perform in a similar fashion within the same ontology. For example as ↑% Patch Risk R ↑% Vulnerability between Patch. However, the Inverse is also proven to be true. A metric scoring high and a metric scoring low within an ontology will have an inverse relationship; i.e. ↑%Application Patch Risk R ↓Safety.

Additionally, by looking at relationships among individual metrics, we can look at combined relationships of metrics within each ontology. If similar reactions to the initial mounted attack occur, this can show each metric within either ontology behave in a similar fashion. When enough of these relationships occur, two ontologies can be shown to be extremely related in security coverage. When this occurs a "super ontology" can be discovered.

The most unique feature of these "super ontology" is the adaptive nature in which they appear with the appropriate adaptive metrics. As such, these are dynamically created from any particular attack. The model for creation of super ontologies will be presented in subsequent sections of his paper.

These will offer large framework metric scenarios that future security work can select from for security solutions.

Autonomous nature to create these systems is also attempted to be proven within this research. Three criteria must be proven to ensure autonomy can be determined. Temporality is determined if two metrics increase following a mounted attack. This proves related behavior and increase a relationship between two ontology. Ontological relationships that increase in tandem can then show entire ontological relationship increases. Finally, correlation following an attack can be determined from the resulting increase ration from the metrics that make them up. Enough of this occurring can place them into "super ontologies following a threshold being reached. When all three of these conditions occur and a system measures it Table 3 represents the relationship between the metrics within the two ontologies O_1 / O_2 and how they interact with each other.

From our research, every metric included within these ontologies behave in a similar manner. The metrics identified as having more interactions within the security model will become the most important metrics per the security needs of the security manager. Adaptive features are realized by how the interactions change with different attacks, different metrics will behave between the ontologies; this will emphasize the importance of different metrics with each attack.

B. Ontological Mathematics

The next premise in the model is that metrics dynamically associate and change to ontological groups based on the contextual threat they are facing. As an example the following metric relation could exist in multiple ontology's or across ontological classes:

$$\text{Patch}\{O,1\} = \uparrow\%r\ R\ \downarrow pl$$

which states that as the decrease in patch level (pl) occurs there is an increase in risk (r) and that this relation could be expressed in fuzzy set theory as the membership function Patch{,}. Quantification of these relations and the rules of the relation are the subject of ongoing work with the ASOMO model.

Based on the degree that these Ontological groupings are related through the similarity matrix, this can help determine how related ontologies are to one another. By determining which ontologies fit best with one another, frameworks can be determined for a best security model. Special autocorrelation is used in this case to show the strength of the relationships. The higher the number that is shown to participate in the relationship, the more correlated the two ontological systems become. Two highly related ontological classes will have metrics respond and thus correlate to a given threat scape or active threat.

In Table 3, some of the relationships between ontologies (O_1 and O_2) and metrics developed as part of this research are demonstrated. Our research shows there are logical arguments that arise from an attack as it affects each metric. For example, consider the metric patch risk (Pr), the introduction of new vulnerabilities when introducing a new patch, has the following interactions; there is an increase in implied patch risk (IPR). Data theft (DT) increases, the risk of an attack occurring with the purpose of bypassing security and stealing data. Correctness (Corr) decreases, the state of the system away from being "fully correct". Reliability

(Rel) decreases, the time in which a failed system is restored. Regularity (Reg) decreases, a state of strictly enforced security. Finally, Security Score (SS) decreases, value of the security of a network. The metric relations to other metrics are shown with up arrows denoting an increase and down arrows denoting a decrease. It is possible from our research to have the same metrics belong in multiple ontologies which leads to the next section were a model of combining metrics in super ontologies is presented. Such recombination relation is stated in the following equation and the basis for combining ontologies of metrics into super and superior ontologies.

O_1 User End		O_2 Server End	
Patch Risk	↑IPR,↑DT ↓Corr ↓Rel ↓Reg ↓SS	Data theft	↓SS↓ Priv ↓ DI
Availability	↓Rel↓Reg↓SS ↑PR ↑ IPR	Data Loss	↓SS ↑ DT ↓ DI
Reliability	↓Ava ↓Reg ↓SS	Privacy	↓SS ↑ DT ↓ DI ↑ Crit
Correctness	↑Ava↑Reg↑SS ↓PR ↓ IPR↓VPH	Workstations (SBI)	↓SS ↑ DT ↑Crit
Implied Patch Risk	↑PR ↓SS↑DT	Systems (AC)	↓SS ↑ DT ↑Crit
Timer	↓SS↑DT	Remote Endpoint Manageability	↓SS ↓Avia ↑ DT ↑Crit
Criticality	↑IPR↓Corr ↓Rel ↓Re	Data Integrity	↓SS ↓Avia
Security Score	↓SS↑DT	Logging Coverage	↓Audit ↓DI
Regularity	↑IPR↓Corr↓Rel	Vulnerabilities per Host	↓SS ↓Audit ↑ DT ↓SS
Vulnerability scanner coverage	↓Rel ↑ VPH	Auditing and Log Files	↓SS ↓VPH ↑ DT ↓SS

Table 3 Metric Interactions Among Metrics and Ontologies

The relationship to threat in such a dynamically organizing system can be stated as the following:

$$f_n(O_1 O_2\ A) = \{0,..,1\}$$

where:

fn() - is a function considering of O_x to O_2
A - is a threat vector

and fn() returns a degree of relation utilized in the process of dynamic reorganization of ontology's composed of metrics.

C. C. Autonomous Ontological Mapping

Having defined a set of metrics that are candidates for Autonomously Organizing Metric Ontology's (AOSMO), we then examined the mechanisms by which ontology's might self-organize. It was determined for the model that the following were possible scenarios for reorganization:

- spatial attribute correlations
- temporal attribute correlations
- conceptual ontological correlations

In spatial correlations there is a relation such as that found over a network, where the ontological metrics are mapped over a network and have spatial adjacency as

malware may pass through the network and fall under the domain of a spatially correlated AOMO's. This scenario can occur as an attack sweeps through a network, penetrating deeper into the core of the system. The concept is modelled as shown in Figure 2.

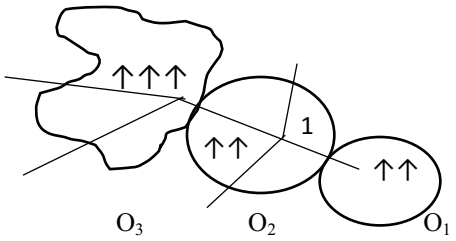


Figure 2. Spatial AOSMO modeling

where:
O_x - are model adjacent ontologies
↑ - model metrics that are increasing as a result of threat as it sweep through
lines - model a network topology

In the above figure, threat orginates at the right in the domain of O_1 and sweeps across the network causing other O_2 etc to respond with heightened metrics.

The mathematical relationship defined in the model is defined as:

$$O_1 \rightarrow O_2 \rightarrow O_3$$

Figure 3. Changes in one ontology imply changes in other ontologies.

stating that increases in metric activity or lack of, implies spatially adjacent AOSMO changes over time as the threat progresses.

Temporal attribute correlation does not have spatial adjacency but rather temporal adjacency. For the purposes of this model, the mathematical relationship is defined as the following:

$$\uparrow O_{1_{t0}} \rightarrow \uparrow O_{2_{t0}}$$

Figure 4. Temporal relation among ontologies

where:
O_x - are temporally related ontologies
↑ - indicates increases in metrics in ontology

This relation implies that at time T_0 an increase one ontologies metrics results in another ontology have a direct relation and increase.

Finally, the conceptual ontological correlation does not look at time or spatial adjacency per se it only looks at the number of metrics in an ontology that increases in response to a specific threat. The relation is defined as the following:

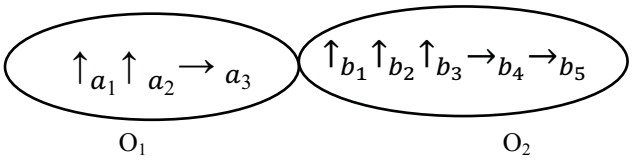


Figure 5. Ontological attribute coorelation

where:
O_x - are different ontology's over a given domain responding to a category of threat
↑ - model metrics that are increasing as a result of a specific threat
→ - indicates metrics that have not changed for metrics a_x and b_x

Work is looking at how to fuzzify the metrics into fuzzy linguistic variables in set theory such as the following example.

$\text{Metric}_x = \{very\ good, good, ok, not\ ok\}$

Finally the AOSMO model defines the concept of a

- superior ontology
- super ontology

Empirical work is planned for the future, but intially the model defines a *superior* ontology to be given as:

$$So_1 \cap So_2 > 50\%$$

Figure 6.. Criteria for creation of a 'super' ontology
where:
S_{ox} - is the set of metrics in onotolgy x

A *super* ontology is defined in the model as

$$So_1 \cap So_2 > 90\%$$

Figure 7. Criteria for creation of a super ontology

where:
S_{ox} - is the set of metrics in onotolgy x

The criteria for a super and superior ontology provide the basis for recombination of ontologies in the AOSMO model. In addition, the numeric criterion can be autonomously adjusted programatically and

implemented in software relatively easily. The relation of super and superior can be stated as the following

$$\uparrow O_1 \cap \uparrow O_2 \rightarrow O^1{}_{Super}{}^{\vee} O_{Superior}$$

Figure 8. General correlation rule

Simply put, this relation states that increases in metric levels for different ontologies 'can' imply the result of a reorganized 'super' or 'superior'
based on the relations previously presented.
Similarities by Fuzzy (move to attributes)
Working on Statistical measures of significant similarity

IV. CONCLUSIONS

This initial work provides the structural model and framework for a system of dynamically and autonomously self-organizing ontology's derived from initial work with defining a set of metrics to operate over. This model used a semi-autonomous approach to correlate relationships based on input metrics. The fuzzy sets allowed for gradient relationships to be formed and evaluated.

Future work can continue to evaluate the definitions and methodology for creating and evaluating the metrics and ontologies used within the model. The proposed ontologies are hypothetical and may or may not be used in an implemented model within a security environment. Therefore, future research can choose which metrics are to be evaluated and organized by the model, and which ontologies can be produced by the data.

The work is initially modeling of a system that could be implemented programmatically. Future work needs to empirically evaluate and validate the models concepts and relations. It also needs to examine the methods of traditional correlation as it might relate to the role fuzzy linguistic variables in the model.

REFERENCES:

[1] Kun Sun; Jajodia, S.; Li, J.; Yi Cheng; Wei Tang; Singhal, A., "Automatic security analysis using security metrics," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011* , vol., no., pp.1207-1212, 7-10 Nov. 2011

[2] Savola, R.M.; Ahola, J., "Towards remote security monitoring in cloud services utilizing security metrics," in Application of Information and Communication Technologies (AICT), 2012 6th International Conference on , vol., no., pp.1-7, 17-19 Oct. 2012

[3] Lingyu Wang, S. Jajodia, A. Singhal, Pengsu Cheng and S. Noel, "k-Zero Day Safety: A Network Security Metric for Measuring the Risk of Unknown Vulnerabilities," in IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 1, pp. 30-44, Jan.-Feb. 2014.

[4] Gregory Vert, Jean Gourd, S.S. Iyengar, 'Application of Context to Fast Contextually Based Spatial Authentication Utilizing the Spicule and Spatial Autocorrelation"

[5] Zonouz, S.A.; Berthier, R.; Khurana, H.; Sanders, W.H.; Yardley, T., "Seclius: An Information Flow-Based, Consequence-Centric Security Metric," in Parallel and Distributed Systems, IEEE Transactions on , vol.26, no.2, pp.562-573, Feb. 2015

[6] K. R. Sekhar, S Reddy L.S. and U. J. Kameswari. Secure system of attack patterns towards application security metric derivation. International Journal of Computer Applications 53(1), 2012.

[7] Jonsson, E.; Pirzadeh, L., "A framework for security metrics based on operational system attributes," in Security Measurements and Metrics (Metrisec), 2011 Third International Workshop on , vol., no., pp.58-65, 21-21 Sept. 2011

[8] G. Vert and S. Baddelpeli, "Adaptive Security Metrics for Computer Systems ," in Proceedings of the 2006 International Conference on Security & Management, 2006 SAM, pp. 1–5

[9] Vaarandi, R.; Pihelgas, M., "Using Security Logs for Collecting and Reporting Technical Security Metrics," in *Military Communications Conference (MILCOM), 2014 IEEE* , vol., no., pp.294-299, 6-8 Oct. 2014
doi: 10.1109/MILCOM.2014.53

[10] Krautsevich, L.; Martinelli, F.; Yautsiukhin, A., "Formal Analysis of Security Metrics with Defensive Actions," in *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)* , vol., no., pp.458-465, 18-21 Dec. 2013doi: 10.1109/UIC-ATC.2013.59

[11] Gopal, A.; Mukhopadhyay, T.; Krishnan, M.S., "The impact of institutional forces on software metrics programs," in *Software Engineering, IEEE Transactions on* , vol.31, no.8, pp.679-694, Aug. 2005

[12] Vert, G.; Doursat, R.; Nasser, S., "Towards Utilizing Fuzzy Self-Organizing Taxonomies to Identify Attacks on Computer Systems and Adaptively Respond," in Fuzzy Systems, 2006 IEEE International Conference on , vol., no., pp.2216-2222, 0-0 0

Generic Semantics Specification and Processing for Inter-System Information Flow Tracking

Pascal Birnstill*, Christoph Bier*, Paul Wagner† and Jürgen Beyerer*
*Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany
Email: {pascal.birnstill|christoph.bier|juergen.beyerer}@iosb.fraunhofer.de
†Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: paul.wagner@student.kit.edu

***Abstract*—Data usually takes different shapes and appears as files, windows, processes' memory, network connections, etc. Information flow tracking technology keeps an eye on these different representations of a data item. Integrated with a usage control (UC) infrastructure, this allows us to enforce UC requirements on each representation of a protected data item. To enable UC enforcement in distributed settings, we need to be able to track information flows across system boundaries. In this paper we introduce a state-based information flow model for tracking explicit flows between systems equipped with UC technology. We demonstrate the applicability of our approach by means of an instantiation in the field of video surveillance, where systems are increasingly accessed via insecure mobile applications. Based on usage control and inter-system information flow tracking, we show how video data transmitted from a video surveillance server to mobile clients can be protected against illegitimate duplication and redistribution after receipt.**

***Index Terms*—Information flow tracking, explicit flows, information flow semantics specification, distributed usage control, policy enforcement**

I. Introduction

Distributed usage control (DUC) is a generalization of access control that also addresses obligations regarding the future usage of data, particularly in distributed settings [1]. UC policies are typically specified via events. Events are intercepted or observed by so-called *policy enforcement points (PEP)* as illustrated in Fig. 1. PEPs forward events to a *policy decision point (PDP)*, which evaluates them against policies. The PDP replies with an *authorization action*, such as *allow, modify, inhibit*, and *delay*, and triggers *obligations*.

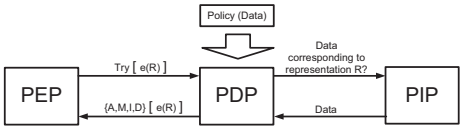


Fig. 1. Generic UC Architecture with Information Flow Tracking

Because data usually comes in different representations – an image can be a pixmap, a file, a leaf in the DOM tree of a website, a Java object, etc.– UC mechanisms have been augmented with information flow tracking technology [2]. One can then specify policies not only for specific fixed representations of a data item, but also on *all* representations of that data item. Policies then do not need to rely on events, but can forbid specific representations to be created, also in a distributed setting [3]. In other words, information flow tracking answers the question into which representations within the (distributed) system monitored data has been propagated.

In order to perform information flow tracking across different applications, different layers of abstraction of a system or across different systems, a multitude of PEPs, each observing an individual set of information flow-relevant events, has to be integrated into the information flow tracking system. The so-called *policy information point (PIP)* interprets the information flow semantics of events and accordingly keeps track of new representations of data being created and of information flows between representations. By this means, when evaluating an event concerning a container (such as a file, process, or window), the PDP can ask the PIP whether this container is a representation of a protected data item, for which a policy must be enforced (cf. Fig. 1).

This work is also explicitly motivated by the increasing number of mobile apps for accessing video surveillance cameras and systems on the market and by the observation that meanwhile video surveillance is entering highly sensitive areas such as hospitals and nursing facilities. While these apps facilitate the cooperation of control rooms and security personnel on-site, we observe that in comparison to heavily secured control rooms the mobile devices being used fall critically short in terms of security mechanisms for protecting the sensitive data captured by surveillance systems. Obviously, the appearance of leaked surveillance footage showing a patient in an emergency situation on a video sharing portal on the Internet is in the interest of neither the patient nor the hospital. We thus instantiate our approach for protecting video data provided by a video surveillance server against illegitimate duplication and redistribution by mobile clients after receipt.

We address the following problems: We generalize an approach to inter-layer information flow tracking introduced by Lovat [4] to additionally cover inter-system flows so as to enable monitoring of flows of protected data between systems equipped with UC enforcement mechanisms. This approach is suitable for proof-of-concept implementation since it is lightweight. Yet it is prone to over-approximations requiring an extension with monitoring technology of higher precision in future work (cf. VI). Plugging new PEPs into existing

UC infrastructures requires information flow semantics of the intercepted events to be deployed at the PIP. We introduce a generic set of primitives for specifying information flow semantics in a uniform syntax to be used by developers of monitors (PEPs). These primitives are derived from analyses of various scenarios in which information flow tracking has been instantiated for UC, such as [2], [5], [6]. Across system boundaries, information flows have to be handled asynchronously, triggered by different events on the particular machines. For this, we specify a protocol for processing inter-layer and inter-system flows based on our semantics description primitives. We thus facilitate UC enforcement on the granularity of representations in distributed settings.

This work is structured as follows. After explaining the formal information flow model of Harvan and Pretschner [2] in Sec. II, we introduce our information flow semantics primitives in Sec. III. In Sec. IV we extend the model so as to allow uniform processing extension of inter-layer and inter-system information flows. We present an instantiation of our approach for protecting video data streamed to a client on behalf of the originating video surveillance server in Sec. V. Eventually we discuss related work in Sec. VI and conclude in Sec. VII.

II. Information Flow Model

Our approach to information flow modeling is based on works of Harvan and Pretschner [2], [5]. An information flow model is a transition system that captures the flow of data throughout a system. Transitions of the state are triggered by events that are observed by monitors, such as PEPs of a UC infrastructure. A system's information flow tracking component, the PIP, interprets events given information flow semantics provided by monitors when being deployed.

The state of the information flow model comprises three aspects. It reflects which data units are in which container, where a container may be a file, a window in the graphical user interface, an object in a Java virtual machine, a network connection, etc. The state also captures *alias relations* between containers, which express that a container is implicitly updated whenever some other container is updated. This happens, for instance, when processes share memory. Finally, the state comprises different *names* that identify a container, e.g., a file may not only be accessible by its file name, but also by a file handle.

A. Formal Model.

As introduced by Pretschner and Harvan in [2], [5] the formal information flow model is a tuple (D, C, F, Σ, E, R). D is the set of data for which UC policies exist. C is the set of containers in the system. F is the set of names. $\Sigma = (C \to 2^D) \times (C \to 2^C) \times (F \to C)$ is the set of possible states, which consists of the *storage function* $s : C \to 2^D$, the *alias function* $l : C \to 2^C$, and the *naming function* $f : F \to C$. Chains of aliases are addressed using the reflexive transitive closure l^* of the alias function. The initial state of the system is denoted as $\sigma_I \in \Sigma$, where the state of the storage function s is given by the initial representation of a data item a UC policy refers to. *Events* E are observed actions that trigger changes of the storage function s, the alias function l, or the naming function f. These changes are described in a (deterministic) transition relation $R \subseteq \Sigma \times E \times \Sigma$. We describe updates to the functions s, l, and f using a notation introduced in [2].

We describe updates to the functions s, l, and f using a notation introduced in [2]: Let $m : S \to T$ be any mapping and $x \in X \subseteq S$ a variable. Then $m[x \leftarrow expr]_{x \in X} = m'$ with $m' : S \to T$ is defined as

$$m'(y) = \begin{cases} expr & if\ y \in X \\ m(y) & otherwise. \end{cases}$$

III. Generic Primitives for Information Flow Semantics

For any PEP, R is specified in an *information flow semantics*, which the PEP deploys on the PIP when being added to a UC infrastructure. For each event intercepted by a PEP, an information flow semantics specifies the state changes of the functions s, l, and f using generic primitives that we introduce in the following. When processing an event according to an information flow semantics (e.g., Listing 2), the PIP picks the action description for the event, converts event parameters in order to match the signatures of the contained semantics primitives (i.e., it implicitly applies f or s on a given parameter: $F \xrightarrow{f} C \xrightarrow{s} D$), and finally modifies its state according to the given primitives.

A. Primitives for Updating the Storage Function

The storage function keeps track of representations, i.e., mappings between data units and containers. We employ it for modeling the actual information flows.

$$\begin{aligned} flow(container\ c,\ data\ \{d_i\}_{1 \le i \le n \in \mathbb{N}}) : \\ s[c \leftarrow s(c) \cup \{d_i\}] \end{aligned} \tag{1}$$

The $flow$ primitive (cf. Eq. 1) indicates an information flow of a set of data units $\{d_i\}_{1 \le i \le n \in \mathbb{N}}$ into the container c. This primitive is used to model that a process creates a new file, a child process, or that a file is copied. Data will then also flow into containers of processes that have a read handle on this file.

$$\begin{aligned} flow\_to\_rtc(container\ c,\ data\ \{d_i\}_{1 \le i \le n \in \mathbb{N}}) : \\ \forall t \in l^*(c) : s[t \leftarrow\ s(t) \cup \{d_i\}] \end{aligned} \tag{2}$$

The $flow\_to\_rtc$ primitive (cf. Eq. 2) models a flow into containers of the reflexive transitive closure $l^*(c)$ of container c. It is used for processes reading from a file, writing to a file, or getting data from the system clipboard.

$$\begin{aligned} clear(container\ c) : \\ s[c \leftarrow \varnothing] \end{aligned} \tag{3}$$

We employ the $clear$ (cf. Eq. 3) primitive whenever a container is deleted, such as when deleting a file, closing a window, killing a process, etc.

B. Primitives for Updating the Alias Function

The alias function maintains relations between containers that lead to implicit flows. Whenever data items flow to container c_{from}, they also flow into the aliased container c_{to}.

$$create\_alias(container\ c_{from},\ container\ c_{to}) : \\ l[c_{from} \leftarrow l(c_{from}) \cup c_{to}] \tag{4}$$

The primitive $create\_alias$ shown in Eq. 4 adds an unidirectional alias from container c_{from} to container c_{to} to the alias function of c_{from}. We use unidirectional aliases for memory-mapped file I/O, if a process has read-only access to the file (cf. `mmap` system call on POSIX-compliant UNIX and Linux systems).

$$create\_bidir\_alias(container\ c_{from},\ container\ c_{to}) : \\ l[c_{from} \leftarrow l(c_{from}) \cup c_{to}], \\ l[c_{to} \leftarrow l(c_{to}) \cup c_{from}] \tag{5}$$

We add bidirectional aliases using the primitive $create\_bidir\_alias$ (cf. Eq. 5). Examples to be modeled with bidirectional aliases include creating a new window, or a process having read and write access to a file.

$$rm\_alias\_locally(container\ c_{from},\ container\ c_{to}) : \\ l[c_{from} \leftarrow l(c_{from}) \setminus c_{to}] \tag{6}$$

The primitive $rm\_alias\_locally$ removes an unidirectional alias from c_{from} to c_{to}, e.g., aliases added using the primitive $create\_alias$ (cf. Eq. 4).

$$rm\_alias\_globally(container\ c_{to}) : \\ \forall c \in C : l[c \leftarrow l(c) \setminus c_{to}] \tag{7}$$

In some cases we also need to remove an unidirectional alias from all containers in C, e.g., in case c is a file, which is deleted. For this, we employ the primitive $rm\_alias\_globally$ as shown in Eq. 7.

$$rm\_bidir\_alias\_locally(container\ c_{from},\ container\ c_{to}) : \\ l[c_{from} \leftarrow l(c_{from}) \setminus c_{to}], \\ l[c_{to} \leftarrow l(c_{to}) \setminus c_{from}] \tag{8}$$

Bidirectional aliases as added using the primitive $create\_bidir\_alias$ (cf. Eq. 5) are removed using the primitive $rm\_bidir\_alias\_locally$ as shown in Eq. 8.

$$clear\_aliases(container\ c) : \\ l[c \leftarrow \varnothing] \tag{9}$$

$clear\_aliases$ removes all aliases with the given container as source from the state of the alias function (cf. Eq. 9), e.g., to clean up if a container is deleted.

C. Primitives for Updating the Naming Function

The naming function maps different names to the same container, e.g., files can be addressed via file names and also via file handles or hard links; in the Windows operating system, we can identify a window via a window handle and also via a window name.

$$add\_naming(naming\ n,\ container\ c) : \\ f[n \leftarrow c] \tag{10}$$

A new name n for a container c is added using the primitive add_naming (cf. Eq. 10) and removed via rm_naming:

$$rm\_naming(naming\ n) : \\ f[n \leftarrow nil] \tag{11}$$

IV. Inter-Layer and Inter-System Flows

So far, our primitives do not capture *inter-layer* and *inter-system* information flows. When using the term *inter-layer*, we refer to flows between different layers of abstraction, e.g., between an application and the operating system. *Inter-system* flows take place whenever data is exchanged between systems over a network connection. We introduce an information flow model extension for monitoring such flows, which requires that an event indicating an *incoming* flow is matched to a preceding *outgoing* event on another system or layer of abstraction.

A. Extended Information Flow Model

As an example, consider the transfer of video data from a streaming server to a client. Assume further that both, server and client, are equipped with PEPs that are capable of intercepting outgoing respectively incoming events as well as with local UC infrastructures. The server side PEP observes an outgoing event indicating a flow from a local container to another container representing the network connection to the client. When receiving data of the video stream via this connection, the client side PEP observes a related incoming event. Finally, when either the client disconnects from the video stream or the server closes the connection, a third event is observed, which terminates the flow. Initially, these events are independent from the perspective of both PIPs. Detecting an inter-system flow requires that both events are interpreted at both PIPs requiring according *remote* information flow semantics, which are provided by the respective PEPs and exchanged between PIPs.

Within an information flow semantics a so-called *scope* specification indicates that an event is related to an event on another system (or layer of abstraction). The particular events are matched to a scope by means of a *scope name* parameter, which is a label for a flow mutually known by two systems (or layers of abstraction). We thus extend the information flow model with a set of scopes $SCOPE$, and the state with the following two mappings: The *intermediate container function* $\iota : SCOPE \rightarrow C$ maps each scope to an intermediate container $c_\iota \in C$. The *scope state function* $\varsigma : SCOPE \rightarrow \{\text{ACTIVATED}, \text{DEACTIVATED}\}$ indicates currently open scopes. Intermediate containers of different systems are

distinct containers, which are mapped on each other by means of scopes and virtually represent the connection. Each event belongs to at most one inter-layer (XLAYER) or inter-system (XSYSTEM) scope. In the initial state σ_I of the system there is one intermediate container c_ι for each scope ι and $\varsigma(sc)$ is DEACTIVATED for all $sc \in SCOPE$. Three attributes of a scope define how the model state is modified when processing an according event:

$$X_{SCOPE} : \Sigma \times E \to SCOPE \times BEHAVIOR \times DELIMITER \times INTER$$
$$DELIMITER = \{\text{OPEN}, \text{CLOSE}, \text{NONE}\}$$
$$BEHAVIOR = \{\text{IN}, \text{OUT}, \text{INTRA}\}$$
$$INTER = \{\text{XLAYER}, \text{XSYSTEM}\}$$

The $DELIMITER$ of a scope describes whether an event indicates a new XLAYER or XSYSTEM flow. The delimiter OPEN changes the state of the scope within which the event is processed to ACTIVATED. The $BEHAVIOR$ describes whether the event indicates an outgoing flow to (OUT), or an incoming flow (IN) from another system or layer of abstraction. The $BEHAVIOR$ of a scope affects the processing of semantics primitives when handling XLAYER/XSYSTEM flows as will be described in Sec. IV-C (INTRA is the default behavior, i.e., a flow within a layer of abstraction, which does not affect the interpretation of primitives). $INTER$ differentiates between XSYSTEM and XLAYER flows.

B. Selecting the Appropriate Scope Semantics for an Event

For each event type a PEP's information flow semantics contains *action descriptions*, which specify its interpretation in terms of information flow using semantics primitives (cf. Sec. III). An action description also includes an ordered list of all scope specifications that possibly apply for this event type.The event notification only contains the scope (as a name-value pair, where the value is the scope itself). When processing an event, the PIP needs to check, in the given order of the action description, which scope specification is applicable. For each scope specification, the PIP evaluates the following three conditions:

1) Does the scope name in the scope specification match the name of a parameter provided in the parameter list of the event notification?
2) If $DELIMITER =$ OPEN in the scope specification: scope deactivated?
3) If $DELIMITER =$ NONE or $DELIMITER =$ CLOSE: scope activated?

If only one of the conditions is not fulfilled, the respective scope is skipped. The ordered list is processed until the matching scope specification X_{SCOPE} is found.

C. Scope Processing

The transition relation R is modified when processing a scope specification. Algorithm 1 describes how R is modified to obtain R_{mod}, i.e., the transition relation for XLAYER or XSYSTEM flows. $R[left \overset{\text{subst.}}{\Longleftarrow} right]$ denotes that the term of R on the left is substituted with the term on the right in R_{mod}. If the delimiter of the scope equals OPEN, the scope is activated (cf. line 6); if the delimiter equals CLOSE, the scope is deactivated after handling the event (cf. line 17). In between (cf. line 8 ff.), depending on the scope's behavior, either the left argument (target) (cf. line 12 ff.) or the right argument (source) of the storage function primitives $flow$ or $flow\_to\_rtc$ is substituted with the scope's intermediate container. R_{mod} is then applied on the state σ (cf. line 16). In case of an XSYSTEM flow, the PIP needs to enable its remote counterpart to process the given event. As described in Sec. V-A this is achieved by forwarding a remote information flow semantics to the remote PIP.

Algorithm 1 Processing an XSYSTEM scope

```
1:  procedure R_inter(σ, e)
2:      (scope, behav, delim, inter) ⟵ X_SCOPE(σ, e)
3:      if scope ≠ ∅ then
4:          ic ⟵ ι(scope)
5:          R_mod ⟵ R
6:          if delim = OPEN then
7:              σ ⟵ ς[scope ← ACTIVATED]
8:          if behav = OUT then
9:              R_mod ⟵ R_mod[s[c ← s(c) ∪ {d_i}]
                ⟸(subst.) s[ic ← s(ic) ∪ {d_i}]]
10:             R_mod ⟵ R_mod[∀t ∈ l(c) : s[t ← s(t) ∪ {d_i}]
                ⟸(subst.) ∀t ∈ l(ic) : s[t ← s(t) ∪ {d_i}]]
11:             R_mod ⟵ R_mod[∀t ∈ l*(c) : s[t ← s(t) ∪ {d_i}]
                ⟸(subst.) ∀t ∈ l*(ic) : s[t ← s(t) ∪ {d_i}]]
12:         if behav = IN then
13:             R_mod ⟵ R_mod[s[c ← s(c) ∪ {d_i}]
                ⟸(subst.) s[c ← s(c) ∪ s(ic)]]
14:             R_mod ⟵ R_mod[∀t ∈ l(c) : s[t ← s(t) ∪ {d_i}]
                ⟸(subst.) ∀t ∈ l(c) : s[t ← s(t) ∪ s(ic)]]
15:             R_mod ⟵ R_mod[∀t ∈ l*(c) : s[t ← s(t) ∪ {d_i}]
                ⟸(subst.) ∀t ∈ l*(c) : s[t ← s(t) ∪ s(ic)]]
16:         σ ⟵ R_mod(σ, e)
17:         if delim = CLOSE then
18:             σ ⟵ ς[scope ← DEACTIVATED]
19:             σ ⟵ s[ic ← ∅]
20:     else
21:         σ ⟵ R(σ, e)
22:     return σ
```

V. Instantiation

We implemented XSYSTEM information flow tracking for a scenario concerning video data provided by a streaming server. It enables us to enforce the UC requirement of preventing redistribution of the video data after receipt on client systems, such as mobile apps for video surveillance systems as mentioned earlier. For this, we need to (i) deploy an according policy at the UC infrastructure of the client, (ii) track the flow of video data from the server to the client, and (iii) monitor the video data at the client so as to inhibit further representations of the data to be created. We achieve (i) and (ii) in the following protocol steps:

1) Intercept an event signaling the *outgoing* data at the server side PEP
2) Evaluate the event against an according policy at the server side PDP
3) Deploy a policy for the data at the client side PDP
4) Process the event at the server side PIP

5) Create a new representation of the video data at the client side PIP
6) Process the outgoing event also at the client side PIP
7) Intercept an event signaling the *incoming* data at the client side PEP
8) Evaluate the event at the client side PDP
9) Process the event at the client side PIP
10) Intercept an event signaling *close* of the connection at the server side PEP
11) Process the event at the server side PIP
12) Process the close event also at the client side PIP

We explain the details of the protocol steps by means of Fig. 2, where an additional component, a system's *policy management point (PMP)*, takes care of policy shipment and deployment.

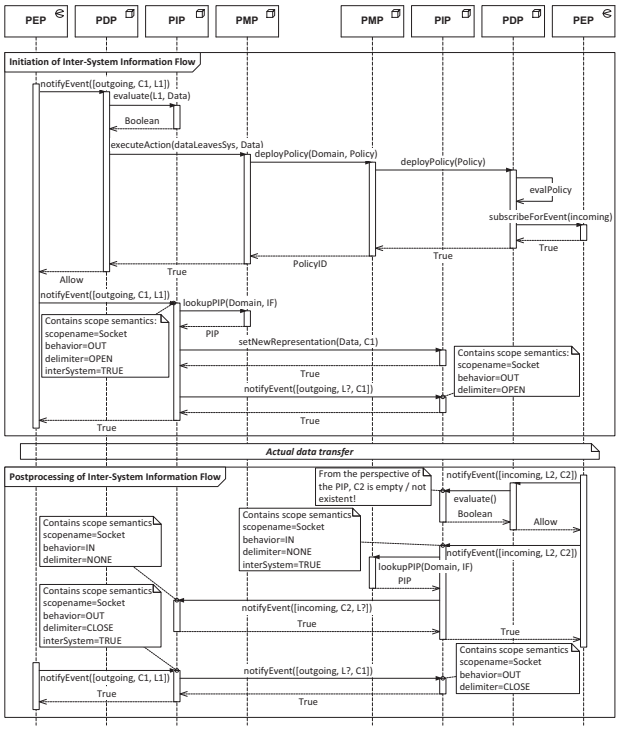


Fig. 2. Inter-System Information Flow

A. Inter-System Information Flow Tracking

Video streaming is triggered by a request from the client, which is intercepted on the server side. The according *outgoing* event triggers the steps 1 to 6 (cf. Listing 1). It indicates

```
<event action="outgoing" timestamp="2015−05−30T09:30:10">
  <parameter name="network" value="192.168.0.2:80;192.168.0.1:49152"> <!−− C1 −−>
  <parameter name="process" value="2a26af9d−f565−4775−87b5−8eb1fb987ad5"> <!−− L1 −−>
  <parameter name="currentscope" value="192.168.0.2:80;192.168.0.1:49152">
</event>
```

Listing 1. Outgoing Event at Server Side

an outgoing flow from the local container *L1*, i.e., the actual server process providing the video stream, to a container *C1* representing the network connection to the client from the perspective of the server.

In step 2, a policy deployed at the server side PDP grants access to the video stream under the condition that a policy for protecting the requested video data is deployed at the client side (step 3): Both policies refer to the video data by means of a unique dataID $data$, which represents the video data within PIPs. Thus, when evaluating the *outgoing* event concerning the local container *L1* against the local policy, the server side PDP queries the local PIP whether *L1* contains $data$ (cf. *evaluate*-call to the server side PIP in Fig. 2). As the PIP returns *true*, the policy matches the outgoing event and evaluates to *allow* under the condition that the policy deployment at the client is successful. This policy demands that no further representations of the protected video data must be created, which includes that it must not be saved and that the screen must not be captured while the video data is accessed.

In step 4 the outgoing event is handled by the server side PIP. The PIP holds a *local semantics* and a *remote semantics* for this event type. The local semantics for the outgoing event is shown in Listing 2. The scope attribute

```
<ifsemantics>
  <params>
    <param name="network" type="CONTAINER"/>
    <param name="process" type="CONTAINER"/>
  </params>
  <actions>
    <action name="outgoing">
      <scope behavior="OUT" delimiter="OPEN" interSystem="TRUE">currentscope</scope>
      <operation name="SF_FLOW">
        <left>
          <operand>network</operand> <!−− C1 −−>
        </left>
        <right>
          <operand>process</operand> <!−− L1 −−>
        </right>
      </operation>
    </action>
    <action name="close">
      <scope behavior="OUT" delimiter="CLOSE" interSystem="TRUE">currentscope</scope>
      <operation name="SF_CLEAR">
        <left>
          <operand>network</operand> <!−− C1 −−>
        </left>
        <right>
        </right>
      </operation>
    </action>
  </actions>
</ifsemantics>
```

Listing 2. Local Semantics of Outgoing and Close Event

$interSystem$ = TRUE in the *outgoing* action description is equivalent to $inter$ = XSYSTEM in the formal model and activates an XSYSTEM scope. The action description indicates a flow from the local container *L1* into the network container *C1*. Due to $behavior$ = OUT of the scope, *C1* is substituted by the scope's *intermediate container* at the server side PIP: As the PIP knows that *L1* contains $data$, it models this flow by mapping $data$ to the intermediate container.

The scope specification in the semantics also triggers the server side PIP to signal the upcoming data transfer to the client side PIP. So far, the client side PIP neither knows that this data exists nor that the client requested it. Step 5 takes care of the first part: The server side PIP creates a new representation of the data at the client side PIP, i.e., we add an initial mapping between the dataID $data$ of the video data and the remote network container *C1* to the client side information flow model. The server side PIP then forwards the event to the client side PIP. In case the remote semantics for this event has not yet been deployed at the client side PIP, it is attached to this notification (cf. Listing 3).

```
<ifsemantics>
  <params>
    <param name="network" type="CONTAINER"/>
    <param name="process" type="CONTAINER"/>
  </params>
  <actions>
    <action name="outgoing">
      <scope behavior="OUT" delimiter="OPEN">currentscope</scope>
      <operation name="SF_FLOW">
        <left>
          <operand>process</operand> <!--L?-->
        </left>
        <right>
          <operand>network</operand> <!--C1-->
        </right>
      </operation>
    </action>
    <action name="close">
      <scope behavior="OUT" delimiter="CLOSE">currentscope</scope>
      <operation name="SF_CLEAR">
        <left>
          <operand>network</operand> <!-- C1 -->
        </left>
        <right>
        </right>
      </operation>
    </action>
  </actions>
</ifsemantics>
```

Listing 3. Remote Semantics of Outgoing and Close Event

In step 6, the client side PIP processes the outgoing event from the server side given the remote semantics (cf. Sec. IV-B). Due to $delimiter$ = OPEN the client side PIP also creates a new scope. The semantics indicates an information flow from the network container *C1* into the container *L?*, which is a wildcard for an unknown container that receives the flow at the client (the local container at the server included in the event is ignored at the client). According to $behavior$ = OUT of the scope, the client PIP replaces *L?* with the scope's *intermediate container* (cf. Sec. IV-C). Together with the fact that *C1* contains $data$, we obtain a flow of $data$ from *C1* into the *intermediate container* at the client side. After this step, the server starts sending video data to the client.

Steps 7 to 9 are triggered by an *incoming* event intercepted by the client side PEP when receiving data over the network connection with the server (cf. Listing 4). The *incoming* event refers to the same *scope* as the *outgoing* event. It indicates a flow from a network container *C2* representing the network connection from the perspective of the client side PEP into a local container *L2*, i.e., the process accessing the video stream. The client side PDP evaluates this event against the policy that has been deployed in step 3. This requires the PDP to query the PIP whether this flow involves a representation of the protected video data (step 8, cf. *evaluate*-call to the client side PIP in Fig. 2). As *C2* is either empty, i.e., it has been created during a prior connection to the server, or does not yet exist, the PIP returns *false*, and the PDP will *allow* the incoming event.

```
<event action="incoming" timestamp="2015-05-30T09:30:11">
  <parameter name="process" value="16820cec-18c7-49a2-a443-cd94f0fec3e0"> <!--L2-->
  <parameter name="network" value="192.168.0.1:49152;192.168.0.2:80"/> <!-- C2 -->
  <parameter name="currentscope" value="192.168.0.2:80;192.168.0.1:49152">
</event>
```

Listing 4. Incoming Event at Client Side

In step 9, the *incoming* event is processed at the client side PIP, which holds a local semantics for this event type. The semantics is shown in Listing 5. It contains a scope specification with $behavior$ = IN and $delimiter$ = NONE. It further signals a flow from the network container *C2* into the local container *L2*. The $delimiter$ = NONE indicates that the event belongs to an already activated inter-system

```
<ifsemantics>
  <params>
    <param name="process" type="CONTAINER"/>
    <param name="network" type="CONTAINER"/>
    <param name="process_id" type="CONTAINER_NAME"/>
  </params>
  <actions>
    <action name="incoming">
      <scope behavior="IN" delimiter="NONE">currentscope</scope>
      <operation name="SF_FLOW">
        <left>
          <operand>process</operand> <!--L2-->
        </left>
        <right>
          <operand>network</operand> <!--C2-->
        </right>
      </operation>
      <operation name="NF_ADD_NAMING">
        <left>
          <operand>process_id</operand>
        </left>
        <right>
          <operand>process</operand> <!--L2-->
        </right>
      </operation>
    </action>
  </actions>
</ifsemantics>
```

Listing 5. Local Semantics of Incoming Event

scope. Due to $behavior$ = IN, *C2* is replaced by the scope's *intermediate container* within the client side PIP. Together with the state after steps 5 and 6, the client side PIP observes a flow of $data$ from the remote container *C1* via the *intermediate container* into *L2*, i.e., as of now, the PIP knows that *L2* contains the video data $data$, which is protected by our policy. Furthermore, a naming is added to the state of the naming function in order to make *L2* accessible via the PID of the process receiving the video data.

Once the client disconnects from the video stream, the established inter-system state is no longer needed, i.e., we deactivate the scopes and delete the intermediate containers at both PIPs. In our example, the termination of the network connection is observed by the server side PEP (step 10). The according event is processed at the server side PIP (step 11) and forwarded to the client side PIP. In line with Algorithm 1, this event is processed with scope delimiter CLOSE at the server and the client according to the scope specification of the local semantics (cf. Listing 2) respectively the remote semantics deployed in step 5 (cf. Listing 3). As *C1* was replaced by the *intermediate container* at the server in step 4, the event has no effect except for closing the scope locally. Tracking of this XSYSTEM flow terminates after the close event is interpreted at the client side (step 12).

B. Client Side Policy Enforcement

In terms of enforcing our policy to inhibit the redistribution of video data at the client (iii), the PIP is queried each time a user triggers an event indicating an according information flow, e.g., when trying to take a screenshot. The event of taking a screen shot is intercepted by a PEP on the client (Android platform, cf. [7] for further details) and is only allowed if no application in the foreground has access to the video stream protected by our policy. For this, the PIP can be queried using the PIDs of questionable processes (cf. V-A, step 9). For the PID of the application accessing the video stream, the PIP will answer that this container is a representation of the data, for which our policy applies. Accordingly, the event, i.e., the screen shot, is inhibited.

The reliability of distributed UC enforcement and likewise the obtained level of security is based on the following

assumptions: The integrity and the correctness of policies and components of the UC infrastructure is ensured. The infrastructure is up and running and not tampered with, i.e., users do not have administrative privileges on their devices.

VI. Related Work

The subject of this paper is specification and processing of information flow semantics depending on events that are intercepted by UC monitors – including inter-system and inter-layer information flows.

Park and Sandhu [8] introduced the first UC model UCON, which has not been combined with information flow tracking. The distributed usage control (DUC) model proposed by Pretschner et al. in [1] has been extended with information flow tracking in [2], [5] in order to enable the enforcement of policies depending on the state of an information flow model. The aspect of distributed enforcement of UC policies is considered in greater detail in [9], [10], also focusing on efficient PDP-PIP communication.

Our work builds on and extends [2]–[5]. We unify information flow semantics specifications of monitoring components and generalize the information flow model to cope with inter-system flows. Since we were up to a lightweight proof-of-concept implementation we did not yet consider monitoring technology with higher precision such as the following. Lovat et al. [4], [11] proposed approaches to handle implicit flows [12] and to address the issue of over-approximations of such simple taint-based information flow tracking systems, which we do not cover.

Information flows towards operating system resources and in-between processes are addressed by taint-based information flow tracking frameworks such as Panorama [13] and TaintDroid [14]. SeeC [15] also covers inter-system taint propagation. With Neon [16], Zhang et al. provide a virtual machine monitor for tainting and tracking flows on the level of bytes, which does not require the modification of applications and operating systems. Demsky's tool GARM [17] tackles data provenance tracking and policy enforcement across applications and systems via application rewriting.

VII. Conclusion

We described and implemented a generic, extensible, and application-oriented approach for dynamic information flow modeling and processing of explicit flows, also across the boundaries of systems equipped with usage control technology. By this means we can enforce usage control requirements on representations of protected data items on remote systems after the initial access to the data has been granted. In our proof-of-concept implementation we have shown how video footage from a surveillance system can be protected against duplication and redistribution even if it is accessed by a mobile application as being employed more and more frequently for cooperation between control rooms and security personnel on-site (provided that the mobile device is equipped with UC technology, otherwise it would not be granted access in the first place).

Our generic primitives for specifying information flow semantics enable engineers to develop information flow monitors (PEPs), which can easily be plugged into existing usage control infrastructures, and thus facilitates the deployment of information flow tracking technology in evolving scenarios. By means of eliminating the interdependency between event capturing and information flow tracking at development time, the practical application of state-based usage control enforcement based on information flow tracking is improved.

References

[1] A. Pretschner, M. Hilty, and D. A. Basin, "Distributed usage control," *Commun. ACM*, vol. 49, no. 9, pp. 39–44, 2006. [Online]. Available: http://doi.acm.org/10.1145/1151053

[2] M. Harvan and A. Pretschner, "State-based usage control enforcement with data flow tracking using system call interposition," in *Proc. NSS*, 2009, pp. 373–380. [Online]. Available: http://dx.doi.org/10.1109/NSS.2009.51

[3] F. Kelbert and A. Pretschner, "Data usage control enforcement in distributed systems," in *Proc. CODASPY*, 2013, pp. 71–82.

[4] E. Lovat, "Cross-layer data-centric usage control," Dissertation, Technische Universität München, München, Germany, 2015.

[5] A. Pretschner, E. Lovat, and M. Büchler, "Representation-independent data usage control," in *Proc. DPM*, 2011, pp. 122–140.

[6] T. Wüchner and A. Pretschner, "Data loss prevention based on data-driven usage control," in *Proc. ISSRE (IEEE)*, 2012, pp. 151–160. [Online]. Available: http://dx.doi.org/10.1109/ISSRE.2012.10

[7] D. Feth and A. Pretschner, "Flexible data-driven security for android," in *Software Security and Reliability (SERE), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 41–50.

[8] J. Park and R. S. Sandhu, "The ucon_{abc} usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004. [Online]. Available: http://doi.acm.org/10.1145/984334.984339

[9] D. A. Basin, M. Harvan, F. Klaedtke, and E. Zalinescu, "Monitoring data usage in distributed systems," *IEEE Trans. Software Eng.*, vol. 39, no. 10, pp. 1403–1426, 2013. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TSE.2013.18

[10] F. Kelbert and A. Pretschner, "Decentralized distributed data usage control," in *Proc. CANS*, 2014, pp. 353–369.

[11] E. Lovat and F. Kelbert, "Structure matters - A new approach for data flow tracking," in *Proc. SPW (IEEE)*, 2014, pp. 39–43. [Online]. Available: http://dx.doi.org/10.1109/SPW.2014.15

[12] E. Lovat, J. Oudinet, and A. Pretschner, "On quantitative dynamic data flow tracking," in *Proc. CODASPY*, 2014, pp. 211–222. [Online]. Available: http://doi.acm.org/10.1145/2557547.2557551

[13] H. Yin, D. X. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: capturing system-wide information flow for malware detection and analysis," in *Proc. CCS (ACM)*, 2007, pp. 116–127. [Online]. Available: http://doi.acm.org/10.1145/1315245.1315261

[14] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. Comput. Syst.*, vol. 32, no. 2, p. 5, 2014. [Online]. Available: http://doi.acm.org/10.1145/2619091

[15] H. C. Kim, A. D. Keromytis, M. Covington, and R. Sahita, "Capturing information flow with concatenated dynamic taint analysis," in *Proc. ARES*, 2009, pp. 355–362. [Online]. Available: http://dx.doi.org/10.1109/ARES.2009.56

[16] Q. Zhang, J. McCullough, J. Ma, N. Schear, M. Vrable, A. Vahdat, A. C. Snoeren, G. M. Voelker, and S. Savage, "Neon: system support for derived data management," in *Proc. VEE*, 2010, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/1735997.1736008

[17] B. Demsky, "Cross-application data provenance and policy enforcement," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, p. 6, 2011. [Online]. Available: http://doi.acm.org/10.1145/1952982.1952988

Selecting Classification Features for Detection of Mass Emergency Events on Social Media

V. Pekar[1], J. Binner[1], H. Najafi[2], and C. Hale[3]
[1]Business School, University of Birmingham, Birmingham, United Kingdom
[2]Computer Science and Information Systems, University of Wisconsin, River Falls, WI, USA
[3]Electronic Systems Lab, Georgia Tech Research Institute, Dayton, OH, USA

Abstract *The paper addresses the problem of detecting eyewitness reports of mass emergencies on Twitter. This is the first work to conduct a large-scale comparative evaluation of classification features extracted from Twitter posts, using different learning algorithms and datasets representing a broad range of mass emergencies including both natural and technological disasters. We investigate the relative importance of different feature types as well as on the effect of several feature selection methods applied to this problem. Because the task of detecting mass emergencies is characterized by high heterogeneity of the data, our primary focus is on identifying those features that are capable of separating mass emergency reports from other messages, irrespective of the type of the disaster.*

Keywords: text classification, feature selection, social media analysis, disaster management

1 Introduction

Social media data offer a promising possibility to deal with mass emergencies. The present-day ubiquity of mobile devices has meant that during a crisis such as a flood, earthquake or a terrorist attack, social media becomes a primary source of information, publishing eyewitness reports on the events in real-time. This gives an opportunity for emergency services to detect crises at early stages, monitor their development and tackle their consequences more effectively.

The potential of social media analysis for mass emergency management has attracted many Data Mining researchers over the past several years. The problem of detecting eyewitness accounts of emergency events in social media has been primarily approached with text classification methods based on machine learning. Limiting the problem to a narrow domain such as earthquakes or tornados has been shown to produce high classification accuracy (e.g., [3, 4, 8, 11, 12, 22]).

However, mass emergency events differ a lot and a classification method that would cover a wide range of possible disasters would be much more practical. This paper is concerned with the broader task of recognizing emergencies unspecified for a particular type, which could include both natural disasters such as earthquakes, floods and storms, as well as man-made ones such as explosions, collisions and shootings. This is a non-trivial classification problem. Firstly, messages relating to a crisis event include not only actual eyewitness accounts but also those that have to do with official announcements, offers of help, sympathy, criticism, and so on. Olteanu et al. [12] report that of all messages judged to be relevant to one of twenty-six mass emergencies, eyewitness accounts comprise only around 8%. The challenge is therefore in identifying specifically eyewitness reports among messages that talk about largely the same event; this is also a classification problem with a big bias towards the negative class. Secondly, because the automatic classifier is expected to operate on a broad variety of event types, each characterized by its own vocabulary. The data is thus not homogeneous: data instances come from related, but different distributions, and in real-world use cases training data is likely to be insufficiently representative of test data on which the classifier is evaluated.

To address these challenges, we study classification features that can be extracted from Twitter messages, beyond the traditional text-based features, that would be suited specifically to the task at hand. Until now, previous papers on detecting emergency-related messages used their own set of features; a few studies examined their contribution to classifier accuracy, but only within a specific application, often limited to one learning algorithm and one emergency event. In this paper we describe a comparative evaluation of a broad set of features that includes those that were used in previous work as well as those introduced for the first time, conducting experiments on data from 26 different emergency events. We report on features that are robust against data heterogeneity and help achieve better classification accuracy when the classifier is evaluated on data from an emergency event that is different from the events exemplified in the training data.

2 Related Work

There is a considerable body of work on detection of new events in a stream of messages, where the type of the event of interest is not known in advance, and some of these approaches were applied to detecting mass emergency events.

Such methods primarily rely on detecting "bursty" keywords [10], i.e. keywords whose frequency increases sharply within a short time window, or bursty message clusters [16]. However, bursty keywords are known to be related not only to new events, but also recurring events and even non-events. To separate them, Becker et al. [2] used a domain-independent text classifier, before applying keyword burstiness techniques.

Domain-specific methods generally have a greater accuracy than domain-independent ones, and previous work specifically on emergency event detection was concerned with developing domain text classifiers based on machine learning and operating on features extracted from the entire message. Most of this work dealt with specific types of crises such as earthquakes [3, 21, 22, 23], bushfires [14], tornados [4, 8], and landslides [11]. Only a few studies developed classifiers that would be applied to more than one type of disasters: Verma et al. [20] evaluated their method on three different types of crises, while Ashktorab et al. [1] on five.

Classification features typically include unigrams (e.g., [1, 15]), bigrams [20, 22], message length [11, 15], part-of-speech tags [4, 19], VerbNet categories [4], the proportion of words that are present in a pre-defined vocabulary [11], whether place names are present [11], hashtags [4, 22], if the message is a retweet [4, 22] or a reply [2]. Verma at al. [20] looked at the contribution of three other kinds of features: if the language of the message objective or subjective, if the register is formal or not, if the text is a first-person report or not.

Any direct comparison between previous approaches is difficult, because they used different experimental datasets, different classification algorithms, and the classification tasks were somewhat different. For example, Imran et al. [5] classified messages into "informative" and "non-informative", Ashktorab et al. [1] into those that report damage and those that do not, Verma et al. [20] into those that are related to situational awareness and those that are not.

3 Classification features

In our evaluation we include the following types of features (examples are shown in parentheses):

Lexical:

`Unigrams`: whitespace-separated word tokens (nominal: *please, help, fire*).

`Bigrams`: token sequences with the length of two (nominal: *was_scary, we_complained*).

`NumberOfUnigrams`: the length of the messages, measured in unigrams. Sakaki et al. [14] found that it was a useful class predictor, as in their data eye-witness accounts tended to be short messages (continuous).

Grammatical:

`Verbs&Nouns`: only word tokens that are tagged as verbs and nouns. The intuition behind these features is that events and their participants are usually described with verbs and nouns, and thus events can be more accurately classified by focusing on verbs and nouns found in the message (nominal: *construction, floor, stuck*).

`PartOfSpeechTags`: separate features are created from part-of-speech (PoS) categories, as assigned by a PoS tagger, the reasoning being that the greater incidence of specific parts of speech (e.g., verbs and nouns) may be more indicative of an eye-witness report (nominal: *NNS, JJ, VBD*).

Semantic:

`VerbNetCategories`: VerbNet [6] is a lexical resource encoding English verbs and different semantic information on them, including their semantic categories. Following Imran et al. [4], for each verb found in a tweet, we add a feature corresponding to its VerbNet category in order to generalize the meaning of specific verbs (nominal: *complain-37.8, get-13.5.1*).

`EMTCategories`: Emergency Management Terms [19] is a lexicon containing around 7,000 words and expressions semi-automatically extracted from Twitter messages on different public emergencies. Each item in the lexicon is associated with a semantic category such as "Caution and Advice", "Injured People", "Infrastructure damage". We detect EM terms in the tweets and use their category labels as features (nominal: *T04, T07, O02*).

`NamedEntities`: We map all named entities, as detected and tagged by the PoS tagger, to a category label, and use it as a feature, instead of actual word tokens (nominal).

Stylistic:

`Sentiment`: We process each tweet with a domain-independent sentiment analysis system [13] and create a feature indicating whether the tweet is neutral in terms of sentiment or not; the system detects emoticons and uses them to determine the sentiment of the message (Boolean).

`Personal`: Following Verma et al. [20], we create a feature indicating if the message contains first-person pronouns ("I", "we", "me", "us") or not, expecting that eyewitness accounts of emergencies will be written from a first-person perspective (Boolean).

`All caps`: We create a feature indicating if the tweet contains all-caps words or not, as words spelled all in capital letters are meant to represent shouting, i.e. used when the author wants to attract special attention to the tweet (Boolean).

Twitter metadata:

`Hashtags`: A hashtag is a word or concatenated phrase preceded by the hash symbol, which are used by authors of messages to group tweets on the same topic and indicate important keywords; we create one extra feature for each hashtag found in a tweet (nominal: *#sandy, #haze*).

`ContainsHashtags`: whether or not the tweet contains any hashtags (Boolean).

`Mentions`: A mention is the name of a Twitter account that is included into the message in order to attract that user's attention to the tweet. We hypothesize that in case crises are reported, the tweet would mention the same set of Twitter

accounts (e.g., news agencies, police, or government bodies). We create one feature for each mention found in the tweet (nominal: *@newscaster, @News1130radio*).

`ContainsMentions`: whether or not the message mentions one or several Twitter accounts. Becker et al [2] found that presence of mentions correlates with reports of emergency events (Boolean).

`RetweetCount`: the number of times the message has been retweeted. We anticipate that eyewitness accounts are likely to attract more interest than other tweets and thus would be retweeted more (continuous).

`Reply`: whether the message is a reply to a different message. In accordance with Becker et al.'s findings [2], we expect that eyewitness accounts will not be replies to previous messages (Boolean).

`ContainsURL`: whether the tweet contains a URL. We expect that eyewitness accounts will tend not to mention any previously published information such as external URLs (Boolean).

Prior to training and classification, all features are converted to the continuous values.

4 Feature selection

Feature selection is a common step in machine learning scenarios, and in particular in text classification, where the number of features is usually very large. It is performed in order to eliminate noisy features, minimize overfitting of the classifier to the training data and to improve its efficiency. In supervised settings, i.e., when class membership of instances is known, the *filtering* approach to feature selection is commonly followed. For an overview of feature selection methods used in text classification, see [17].

In the context of tweet classification the filtering approach can be formalized as follows. Let us assume that each tweet $t \in T$ of the training set is represented as a feature vector, consisting of features $f \in F$, and that each t is assigned a class label $c \in C$. For each f, from its distribution across C, a certain function computes its informativeness score $s(f,c)$, specific to each class. From class-specific scores, one can compute its global score by, e.g. averaging local scores of f across classes. The features are then sorted by their informativeness and k top features are selected to represent instances, with k set experimentally. After non-informative features have been removed from the training data, a classifier is learned and evaluated on the test data.

A key decision for feature selection is to choose a function computing $s(f,c)$. Such functions aim to capture the intuition that the best features for a class are the ones that best discriminate between its positive and negative examples. They determine $s(f,c)$ from the distribution of f between c and non-c, attributing greater weights to those f that correlate with c or non-c the most. In the present study we include three such functions widely used in text categorization.

Chi-square. The chi-square (CHI) statistic measures the lack of independence between f and c, and can be used directly as the informativeness score. The chi-square is calculated between the observed frequency of co-occurrence of f and c $fr(f, c)$ and their expected co-occurrence $fr'(f, c)$. First the latter is obtained assuming the f and c co-occur randomly:

$$fr'(f,c) = \frac{fr(f) \cdot fr(c)}{\sum_{m \in F} \sum_{n \in C} fr(m,n)}$$

The chi-square statistic is then calculated as:

$$\chi^2(f,c) = \sum_{m \in \{f, \bar{f}\}} \sum_{n \in \{c, \bar{c}\}} \frac{(fr(m,n) - fr'(m,n))^2}{fr'(m,n)}$$

Information Gain. IG measures the number of bits of information obtained about presence and absence of c by knowing the presence or absence of f. It is calculated as follows:

$$IG(f,c) = \sum_{m \in \{f, \bar{f}\}} \sum_{n \in \{c, \bar{c}\}} p(m,n) \log \frac{p(m,n)}{p(m)p(n)}$$

Information Gain Ratio. IGR is a normalized version of IG, meant to overcome the disadvantage of IG that it grows not only with the increase of dependence between f and c, but also with the increase of the entropy of f. IGR removes this factor by normalizing IG by the entropy of c:

$$GR(f,c) = \frac{IG(f,c)}{-\sum_{n \in \{c, \bar{c}\}} p(n) \log p(n)}$$

5 Experimental setup

5.1 Data

For experimental evaluation we use the labeled part of the CrisisLexT26 dataset [12], which includes tweets on 26 mass emergencies that occurred in 2012 and 2013. The types of emergencies are very diverse and range from terrorist attacks and train derailment to floods and hurricanes. Some examples are Colorado wildfires in 2012, Venezuela refinery explosion in 2012, Boston bombings in 2013.

There are 24,589 labeled tweets in the dataset in total, with 2,193 of them labeled as originating from an eyewitness. The classification task in our experiments consisted of predicting whether a given tweet was an eyewitness report or not.

5.2 Preprocessing

We apply the following preprocessing steps to the data:

Additional metadata. The CrisisLexT26 data contains the Twitter id of the message, its raw content, and its timestamp.

Via Twitter Search API we retrieve additional metadata fields: retweet count, reply, hashtags.

Deduplication. Duplicate tweets were removed by measuring similarity in each pair of tweets using cosine and removing one tweet in pairs where the cosine was higher than 0.99.

Tokenization and part-of-speech tagging. Before processing the text of the message with a PoS tagger, the text was normalized: mentions (e.g., @someone) and URLs removed; sequences of hashtags at the start and end of the message removed; hashtags appearing in the middle of the text were kept, but the hash symbol removed from the hashtags; long non-alphanumeric symbol sequences, which tend to be emoticons, were removed; word tokens consisting of digits were replaced with a unique tag. The normalized text was tokenized and tagged with the PoS tagger in the Pattern library [19].

Sentiment analysis. The original text of the tweet was processed with the sentiment analysis system [13]. The system was used in the SemEval ABSA challenge, where it achieved an F-measure of 0.67 and 0.75 on the two evaluation datasets within the sentiment analysis subtask. The system assigned to an input text a sentiment score between -1.0 (negative) and 1.0 (positive); the score was converted to a Boolean value indicating if the tweet is neutral in terms of sentiment (the score was equal 0) or not.

Stopword removal. The usual stoplist was used to remove stopwords.

5.3 Evaluation Metrics

The accuracy of classification is measured in terms of the traditional measures of precision, recall and F1 measure. Because the data is biased towards the negative class, the evaluation metrics averaged over both classes may be misleading, so we report them only for the positive class, i.e. the eyewitness report class.

6 Results and Discussion

6.1 Impact of Data Heterogeneity

In the first experiment, we examined the extent to which data heterogeneity present in the CrisisLexT2 dataset affects classification accuracy. To that end, we evaluated the classifiers in two scenarios. In the first ("Scenario 1"), the entire dataset was randomly split into a train and a test set, in proportion 1 to 9. This ensured, with a large likelihood, that data on the same crisis will be present in both training and test data, and the feature distribution in the test data will be similar to the one in the train data.

The second scenario ("Scenario 2") was meant to better reflect real-world use cases: the train-test split was done in such a way so that the test data contained tweets only on those crises that were not included into the train data, i.e., simulating the conditions when a crisis needs to be detected before any manually labelled data relating to it are available. Specifically, data on 23 crises were used for training and data on 3 remaining crises were used for testing.

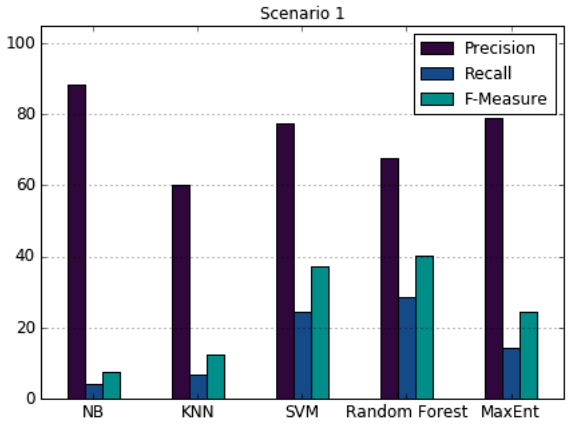


Figure 1. Classifier performance on the full set of features, random train-test split ("Scenario 1").

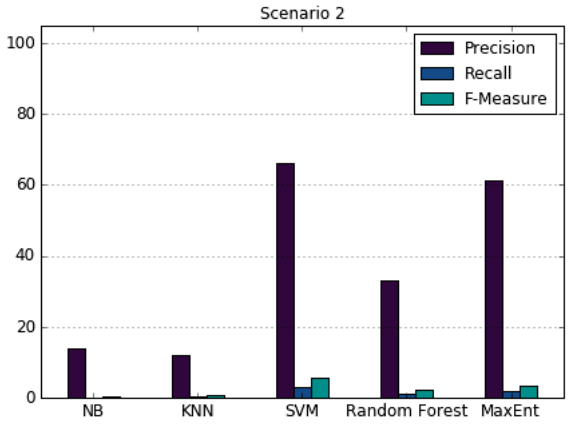


Figure 2. Classifier performance on the full set of features ("Scenario 2").

Training on all the features described in Section 3, we compared the performance of five classifiers – Naive Bayes, k Nearest Neighbors (kNN, k=5), Random Forest, Maximum Entropy (MaxEnt, a.k.a. Logistic regression) and linear Support Vector Machines (SVM), under these two scenarios. The results are shown in Figures 1 and 2

The results show that Scenario 2 is indeed a much harder evaluation task: both precision and recall rates for all the five classifiers drop; the drop is especially big for recall (e.g., for SVM it falls 0.24 from to 0.01). This suggests that, as anticipated, there is more data heterogeneity between different crises than within them. To confirm this, we measured the difference between distributions of features in each train-test split using Jensen-Shannon divergence, a variant of Kullback-Leibler divergence [9]; feature probabilities are obtained via Maximum Likelihood Estimation. We find that the average JS divergence in Scenario 1 is 0.01, while in Scenario 2 it is much higher, at 0.07, the difference is significant based on an independent samples t-test ($p < 0.001$).

In our subsequent experiments, we used the SVM and MaxEnt classifiers, which fared better than the other three.

6.2 Feature types

To measure the relative utility of each type of features, we ran experiments where each feature type was removed from the full set of features and the change in classifier performance was noted. The results for Scenario 1 are shown in Tables 1 (SVM) and 2 (MaxEnt), for Scenario 2 – in Tables 3 (SVM) and 4 (MaxEnt), the tables show the percent changes in F-measure, precision and recall resulting from removing one feature type.

	F-measure	Precision	Recall
Bigrams	-3.39	0.70	-2.91
Hashtags	-3.10	0.16	-2.53
Mentions	-1.51	-1.47	-1.18
ContainsMentions	-0.83	-0.08	-0.63
EMCategories	-0.78	-0.75	-0.58
Sentiment	-0.39	0.70	-0.39
AllCaps	-0.35	-0.71	-0.22
Personal	-0.34	0.05	-0.29
Reply	-0.23	0.54	-0.21
PosTags	-0.20	-0.98	-0.08
ContainsUrl	-0.16	-0.11	-0.12
Verbnet	0.03	-0.15	0.03
ContainsHashtags	0.22	0.10	0.21
VerbsAndNouns	0.61	-1.19	0.70
NumberOfUnigrams	0.94	-0.80	0.99
NamedEntities	3.10	1.14	2.67
RetweetCount	9.05	-0.22	8.51

Table 1. The effects of removing one feature from the feature set, Scenario 1, SVM.

	F-measure	Precision	Recall
PosTags	-2.66	-0.77	-1.79
Hashtags	-2.42	1.01	-1.62
EMCategories	-1.19	1.29	-0.84
ContainsHashtags	-1.14	-0.16	-0.76
Personal	-0.82	-0.72	-0.54
AllCaps	-0.32	-0.74	-0.16
Mentions	-0.28	-0.55	-0.15
Sentiment	-0.17	-1.33	-0.07
Reply	-0.14	0.0	-0.05
ContainsMention	0.04	0.68	0.02
ContainsUrl	0.04	-1.78	0.13
Verbnet	0.07	-0.64	0.08
VerbsAndNouns	0.44	-0.42	0.32
Bigrams	1.84	-4.92	1.54
NumberOfUnigrams	2.50	-1.80	1.87
NamedEntities	3.61	1.40	2.55
RetweetCount	9.50	2.57	7.04

Table 2. The effects of removing one feature from the feature set, Scenario 1, MaxEnt.

For Scenario 1 results, we see that the changes are not very significant, except for `Hashtags`, which contribute a lot to the recall of the classifiers (up to 5 points), `Bigrams`, which help precision for Maxent and recall for SVM, and `RetweetCount` and `NamedEntities`, whose removal leads to improvements in all the three metrics, by up to 9 points. Some features increase precision at the cost of recall (`NumberOfUnigrams`), while others, on the contrary, improve recall at the cost of precision (`HashTags`, `EMCategories`).

For Scenario 2, the changes in F-measure are not high, but differences between specific features in terms of precision and recall are much more noticeable than for Scenario 1. For both classifiers, `Bigrams` are important for precision, the changes are 11.7 points for SVM and 8.2 for MaxEnt, while `EMCategories` and `Personal` help to improve recall. The use of `PosTags` improves all the evaluation metrics for both classifiers. `RetweetCount`, `VerbNet`, `HashTags` and `VerbsAndNouns` produce an adverse effect on all the three metrics, also for both SVM and MaxEnt. A somewhat unexpected observation is that `PosTags` are positively influencing all the metrics, for all classifiers and scenarios. The changes for other features are less consistent between the classifiers.

	F-measure	Precision	Recall
Bigrams	-1.97	-11.70	-1.08
Personal	-0.97	-5.20	-0.55
NamedEntities	-0.26	-2.48	-0.15
ContainsHashtags	-0.25	6.82	-0.17
PosTags	-0.22	-3.44	-0.12
EMCategories	-0.20	2.20	-0.13
Sentiment	-0.18	-2.44	-0.11
ContainsMention	-0.15	-0.05	-0.09
ContainsUrl	-0.02	1.77	-0.04
Reply	0.0	0.0	0.0
Mentions	0.03	3.71	0.01
AllCaps	0.03	-0.85	0.01
Verbnet	0.22	0.44	0.11
Hashtags	0.53	9.39	0.26
RetweetCount	1.25	-3.49	0.75
NumberOfUnigrams	1.3	-5.16	0.76
VerbsAndNouns	1.58	6.46	0.85

Table 3. The effects of removing one feature from the feature set, Scenario 2, SVM.

More generally, it seems that lexical features such as `Bigrams` help to achieve greater precision, while Semantic (e.g., `EMCategories`), Stylistic (e.g., `Personal`) and Twitter-related (e.g., `ContainsHashtags`) ones – greater recall. These characteristics of the features become more prominent in Scenario 2.

	F-measure	Precision	Recall
EMCategories	-1.13	2.24	-0.63
PosTags	-0.97	-38.77	-0.51
ContainsHashtags	-0.68	-2.0	-0.38
Personal	-0.68	5.52	-0.39
ContainsUrl	-0.34	-4.71	-0.19
NamedEntities	-0.31	2.31	-0.19
Sentiment	0.0	0.0	0.0
Mentions	0.0	0.0	0.0
AllCaps	0.14	1.17	0.07
Reply	0.15	1.85	0.07
VerbsAndNouns	0.37	5.27	0.18
ContainsMention	0.40	2.22	0.22
Bigrams	0.62	-8.24	0.36
Verbnet	0.76	2.46	0.41
Hashtags	0.93	2.77	0.50
RetweetCount	0.98	-8.69	0.56
NumberOfUnigrams	1.55	2.45	0.85

Table 4. The effects of removing one feature from the feature set, Scenario 2, MaxEnt.

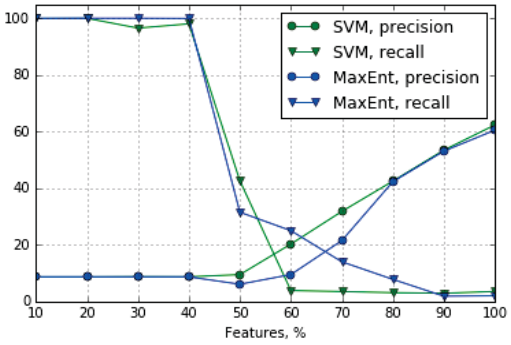


Figure 3. The effect of feature selection based on CHI on precision and recall of SVM and MaxEnt.

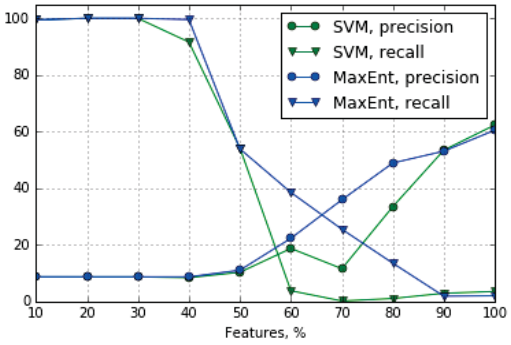


Figure 4. The effect of feature selection based on IG on precision and recall of SVM and MaxEnt.

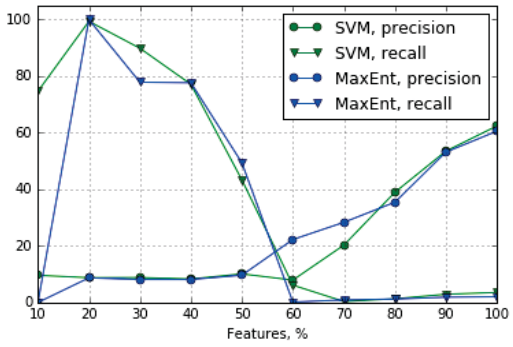


Figure 5. The effect of feature selection based on IGR on precision and recall of SVM and MaxEnt.

6.3 Feature Selection

In the next experiment, we examined the ability of the Chi-Square, Information Gain and Information Gain Ratio to select the most useful classification features. Computing scores for all the features, we experimented with keeping the top 10%, 20%, …, 90% of the most informative features. These results are shown in Figures 3, 4, and 5.

We see that the effect of feature selection is largely similar for CHI, IG and IGR. When features are removed drastically (keeping 40% of features or less), the recall improves all the way to 100%, while precision drops to about 10%. The fewer features are removed, the greater precision and the lower recall, with the best precision achieved when keeping 100% of features, both classifiers and all three feature ranking functions.

7 Conclusion

This is the first work to conduct a large-scale comparative evaluation of classification features extracted from Twitter posts, using different learning algorithms and datasets representing a broad range of mass emergencies including both natural and technological disasters. Our key findings can be summarized as follows. We presented empirical results demonstrating that a machine learning classifier tested on data that represents mass emergency events that were unseen at the training stage suffers a significant performance drop, especially in terms of recall, in comparison to testing on data that represents the same types of emergency events as the train data. We furthermore find that when testing the classifier on unseen event types, lexical features help to achieve better precision, while semantic, stylistic, and features derived from message metadata help improve recall. Finally, we examined several well-known feature selection methods, finding that they all produce a similar effect on the classifier: at aggressive levels of feature selection, they lead to better recall; however, they do not help much with precision.

This work has thus produced results that can inform development of applications for automatic detection of social media posts relating to mass emergencies, with regards to the

choice of features to be used under different use cases. Future research will focus on ways to exploit the described properties of the features: for example, create different feature subsets constituting different "views" on the data within a semi-supervised learning method.

8 References

[1] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining Twitter to inform disaster response. In Proc. of ISCRAM.

[2] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on Twitter. Proc. of ICWSM. 438–441.

[3] Cornelia Caragea, Nathan McNeese, Anuj Jaiswal, Greg Traylor, H. Kim, Prasenjit Mitra, Dinghao Wu, A. Tapia, Lee Giles, Bernard J. Jansen, and others. 2011. Classifying text messages for the Haiti earthquake. In Proc. of ISCRAM.

[4] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, Patrick Meier. 2013. Extracting Information Nuggets from Disaster-Related Messages in Social Media. In Proc. of ISCRAM.

[5] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. AIDR: Artificial intelligence for disaster response. In Proc. of WWW (Companion). IW3C2, 159–162.

[6] Karin Kipper, Anna Korhonen, Neville Ryant and Martha Palmer. Extending VerbNet with Novel Verb Classes. Proceedings of the Fifth International Conference on Language Resources and Evaluation -- LREC'06. May, 2006, Genoa, Italy: 2006.

[7] Rui Li, Kin Hou Lei, Ravi Khadiwala, and KC-C Chang. 2012. Tedas: A Twitter-based event detection and analysis system. In Proc. of ICDE. IEEE, 1273–1276.

[8] Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during hurricane Irene. In Proc. of the Second Workshop on Language in Social Media (LSM '12). Association for Computational Linguistics, Stroudsburg, PA, USA, 27-36.

[9] Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.

[10] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Twitinfo: Aggregating and visualizing microblogs for event exploration. In Proc. of CHI. 227–236.

[11] Aibek Musaev, De Wang, and Calton Pu. 2014. LITMUS: Landslide detection by integrating multiple sources. Proc. of ISCRAM.

[12] Alexandra Olteanu, Sarah Vieweg, Carlos Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In Proceedings of the ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '15). ACM.

[13] Viktor Pekar, Naveed Afzal, and Bernd Bohnet. 2014. UBham: Lexical Resources and Dependency Parsing for Aspect-Based Sentiment Analysis. In Proc. of the Eighth International Workshop on Semantic Evaluation (SemEval 2014).

[14] Robert Power, Bella Robinson, John Colton, Mark Cameron. 2015. A Case Study for Monitoring Fires with Twitter. In Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM'15).

[15] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proc. of WWW. ACM, 851–860.

[16] Vincent Schmidt and Jane Binner. 2011. A Semi-automated Display for Geotagged Text. Proceedings of the 2011 International Conference on Software Engineering Research and Practice.

[17] Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002.

[18] Tom De Smedt and Walter Daelemans. (2012). Pattern for Python. Journal of Machine Learning Research. 13, 2063-2067.

[19] Irina Temnikova, Carlos Castillo, and Sarah Vieweg. 2015. EMTerms 1.0: A Terminological Resource for Crisis Tweets. In Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM'15).

[20] Sudha Verma, Sarah Vieweg, William J. Corvey, Leysia Palen, James H. Martin, Martha Palmer, Aaron Schram, and Kenneth Mark Anderson. 2011. Natural language processing to the rescue? Extracting "Situational Awareness" tweets during mass emergency. In Proc. of ICWSM.

[21] Hiroko Wilensky. 2014. Twitter as a Navigator for Stranded Commuters during the Great East Japan Earthquake, Proceedings of the 11th International ISCRAM Conference.

[22] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. IEEE Intelligent Systems 27, 6, 52–59.

[23] Andrea Zielinski and Ulrich Bügel. 2012. Multilingual Analysis of Twitter News in Support of Mass Emergency Events. In of the 9th International ISCRAM Conference – Vancouver, Canada, April 2012.

PUF based Lightweight Hardware Trust Anchor for Secure Embedded Systems

Kai Fischer[1], **Andreas Mucha**[1], **Erwin Hess**[1], **and Fabian Riess**[1]
[1]Corporate Technology, Siemens AG, Munich, Germany

Abstract— *Storage of cryptographic keys on embedded systems is often one of the weakest points of their security architecture, especially if they do not provide a separate security chip, e.g., for cost reasons. But even devices with a security IC in addition to the main controller often lack in mechanisms to securely pair the insecure system and the security IC and are vulnerable for local attackers. We propose the application of Physical Unclonable Functions (PUFs) to realize a lightweight and secure universal key provision token (PUF-KT) to store different keys on devices without a secure non volatile memory (NVM). For embedded devices with a security IC, PUF-KT is used to cryptographically bind the main controller with the security chip without a need to store authentication credentials on the insecure part of an embedded system. As PUF implementation for these mechanisms, we improved the concept of bistable ring PUFs and developed a flexible delay based PUF named Criss-Cross PUF.*

Keywords: PUF Key Provisioning Token, Trust Anchor, Criss Cross PUF, Physical Unclonable Function, Embedded Security

1. Introduction

In recent years the amount and professionalism of security attacks on embedded systems has increased significantly. One reason is that Internet-of-Things and cyber physical system (CPS) scenarios are driving the interconnection/communication between embedded devices. Another reason is that embedded devices are getting smarter and gain more attraction for attacks. A broad set of security mechanisms, cryptographic protocols and algorithms are required in open scenarios to protect the confidentiality, integrity and authenticity of communication and data at rest.

Cryptographic protocols and algorithms require secret/private keys and, therewith, it is needed to generate, store and manage the necessary keys on embedded devices. Typically, keys are generated based on physical or seeded pseudo-random number generators and are then permanently stored in a protected memory area (non volatile memory (NVM) or a battery powered RAM). Unfortunately, embedded systems often do not provide a secure memory. Separate security ICs that would offer secure key management and memory are typically not available, commonly due to cost limitation requirements of constrained commercial products. But even devices providing a security IC in addition to the main controller lack in mechanisms to control the access to the security IC as no cryptographic binding between the insecure system and the security IC exists.

In this paper, we propose the application of a lightweight and secure hardware trust anchor for embedded devices based on Physical Unclonable Functions (PUFs). This universal key provision token (PUF-KT) covers three important requirements simultaneously. Firstly, as a PUF – ideally – produces device specific random-looking and unpredictable data it relieves from implementing a physical random source for key generation. Secondly, it addresses scenarios in which a secure NVM for key-storage is not available, as the PUF will generate the desired device specific key every time it is activated. The key provision token allows to generate symmetric as well as asymmetric keys, e.g., for encryption of memory resources, device authentication or secure communication. Thirdly, for the class of embedded devices that already have a security IC on board, we apply the PUF-KT to achieve a cryptographic pairing between the micro controller of an embedded system and the security IC and mitigate attacks in which the security IC is used in unintended contexts.

The rest of the paper is organized as follows: In Section 2, we describe our key provision token approach, followed by Section 3 where we outline possible PUF-KT based usage scenarios like a secure pairing mechanism. In Section 4, we introduce the concept of the Criss-Cross PUF as basis of our PUF-KT implementation. We improved the concept of delay based bistable ring PUFs and provide first evaluation results. At the end, we give a conclusion and outlook of further work.

2. A Universal PUF based Key Provision Token

2.1 Related Work

The direct use of PUFs for device authentication has been intensively discussed in recent years, but the outcome of these investigations is rather negative. The main problem of all the known concepts is that the proposed schemes are vulnerable to machine learning attacks. These attacks yield a software clone that is indistinguishable in its challenge-response-behavior from that of the original PUF. For a comprehensive overview on the status of direct PUF based authentication we refer to [1], [2]. One may conclude that

– based on the currently available PUFs – these concepts are not yet mature enough for deployment. On the other hand, these problems are not relevant in the key generation scenario in which the PUF output is never transferred to the "outer world" for direct use. It is used device internal only to generate secret/private keys for usage in cryptographic schemes.

Currently known PUFs cannot be used directly for key provision as they do not show the ideal behavior to produce in a *stable way* data that are *device specific*, *unpredictable* and *random*. In the literature one can find various approaches to derive a secret key from the output of a PUF; we refer especially to [3]–[5] for detailed descriptions of different concepts. Roughly speaking, all the suggested PUF key provision methods are based on the components of a raw PUF, an entropy extractor (*EE*) and a post-processing function (*PP*) (for implementation variants of *EE* and *PP* we refer also to [6]–[8]).

2.2 Our Approach

Our idea of a universal PUF based key provision token was not to develop a PUF based alternative to a secure key storage able to output one fixed secret key in a reproducible way. Instead, our goal was to provide a *universal* and *configurable* approach to produce all types of keys appropriate to enhance the internal security of the embedded system. Also, it should be possible to generate the keys necessary for the support of modern communication security protocols as, e.g., TLS.

Altogether, for these purposes four types of (operational) keys are required:

1) A symmetric key – $K_{P,sym}$ – for permanent usage in the embedded system (e.g., for secure data storage in the external unprotected NVM of the embedded system).
2) A private key – $K_{P,priv}$ – for permanent usage for device-bound digital signatures (e.g., for public-key based certificates).
3) Session specific private keys – $K_{S,priv}$ – for short term usage (e.g., in communication protocols, for key negotiation or as ephemeral key in digital signature schemes).
4) Session specific symmetric keys – $K_{S,sym}$ – for short term usage (e.g., for data encryption, message authentication in communication protocols).

The standard process to derive a cryptographic key from a PUF comprises the following steps:

1) Querying the raw PUF structure and transfer of the PUF output to the entropy extractor.
2) "Stabilizing" the PUF output in *EE* using the helper data specific to the given PUF implementation.
3) "Homogenizing" the *EE* output in a post-processing stage to meet the quality requirements (mainly statistical properties) for cryptographic keys.

We modify and extend this process to produce the four desired key types. Generally, the PUF-KT structure offers two modes to generate symmetric as well as asymmetric keys. The mode controls the number of PUF bits to be generated, the output lengths of *PP* and whether the desired key shall be a permanent or a session key. A configuration vector, which is stored in the internal ROM or in fuses, acts as initial value for a configurable feedback shift register to generate the challenge C_{sym} or C_{priv} dependent on the selected mode. Based on the challenge, *PP* produces a binary master vector, either $K_{M,sym}$ or $K_{M,priv}$ of the appropriate lengths for secret and public key cryptography in a reproducible way. We consider the two keys $K_{M,sym}$ and $K_{M,priv}$ as the symmetric and the asymmetric PUF-KT internal master keys intrinsically related to the specific PUF implementation under the fixed start configuration.

Fig. 1 shows the architecture of the universal PUF-KT and the flow of data.

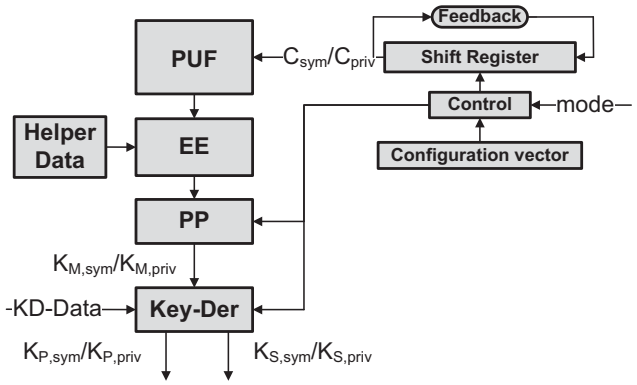


Fig. 1: Universal PUF based key provision token - Architecture

For derivation of the four desired key types $K_{P,sym}$, $K_{P,priv}$, $K_{S,sym}$ and $K_{S,priv}$ we use a key derivation function (Key-Der) taking – respectively – $K_{M,sym}$ or $K_{M,priv}$ as master keys and some key derivation data (KD-Data) as additional input. These data may be generated internally or be fed in from outside and allows us to generate different application specific keys. For permanent keys, static data is required as input for key derivation, whereas for session keys volatile data shall be applied (e.g. counter values, random numbers).

The asymmetric cryptographic schemes we have in mind to work with the keys $K_{P,priv}$ and $K_{S,priv}$ are elliptic curve based schemes like ECDSA and ECDH. From a cryptographic point of view elliptic curve cryptography (ECC) is currently the most attractive public-key system regarding performance and "security per bits". For a detailed reference on elliptic curves and ECC we refer to [9]. We mention here only some important properties that simplify essentially the use of ECC as PUF-KT asymmetric scheme compared to

RSA:

- ECC works with fixed system parameters. These are the underlying finite field $\mathbb{F}$, the equation defining the elliptic curve E over $\mathbb{F}$ and a fixed chosen point P on E – the base point of the system.
- There is an operation – point multiplication – that adjoins to any integer $[k]$ and the point P a new point $Q := [k] \cdot P$ on E.
- The public keys are derived from the private keys via point multiplication. Given $K_{M,priv}$ and a point P, we derive the asymmetric permanent and session keys $K_{P|S,priv}$ by applying the key derivation function $Key - Der$. The associated public keys are generated by a point multiplication $K_{P|S,pub} = [K_{P|S,priv}] \cdot P$ where the binary private keys have to be applied as positive integers.

In contrast to that, the RSA scheme is not well suited for interplay with the PUF-KT generated private master key $K_{M,priv}$. Actually, it is rather difficult and inefficient to establish a RSA public key system on the PUF-KT approach. $K_{M,priv}$ may be used to derive two secret, randomly looking and fixed seeds for the generation of the two primes p and q whose product constitute the RSA module $N = p \cdot q$. To complete the RSA setup one chooses in the next step the intended public key e. To obtain the adjoined private key d for the RSA system one has to solve the equation

$$e \cdot d \equiv 1 \mod N.$$

Hence, in the case of the RSA scheme the secret information $K_{M,priv}$ derived from the intrinsic PUF secret is only a starting point for the complete RSA parameter construction procedure in contrast to ECC.

2.3 Lifetime properties of the PUF-KT generated keys

The keys generated by the universal PUF-KT have different lifetime properties. All keys get lost after power-off of the embedded system as no secure NVM is claimed to be available. The keys $K_{P,sym}$ and $K_{P,priv}$ are reproduced based on static (unprotected) NVM data by PUF-KT after system restart. This does not hold for the session specific keys $K_{S,sym}$ and $K_{S,priv}$ that depend on volatile data.

These different lifetime properties of the keys yield in specific security implications: Compromise of a permanent key does not affect the PUF-KT internal master keys and the derived session keys (under the assumption a strong key derivation functions is used). The same holds for compromise of session keys vice versa. If one the two master keys $K_{M,sym}$ and $K_{M,priv}$ is broken, a security application using the permanent keys gets insecure. This is different for the session specific keys. If the volatile data used for key derivation are lost, it is impossible to regain these keys. Hence, usage of session keys derived from the master keys supports *forward security*. However, for future use of the master keys it is required to re-configure the feedback shift register to generate different PUF challenges C_{sym} or C_{priv} and therewith fresh master keys $K_{M,sym}$ and $K_{M,priv}$ if one the two master keys is broken.

3. Applications of the PUF-KT

This section gives two important applications that make use of PUF key token.

3.1 External Memory Protection

Crucial security problems in any embedded device scenario are the authenticity and confidentiality of data or programs stored in the external EEPROM/Flash and RAM memories. If the embedded device is not physically protected – this is the usual case – an attacker can easily access and/or change these data. Confidentiality of data can be protected using symmetric cryptographic algorithms ENC with the key $K_{P,sym}$. The data DAT to be stored in external memory is encrypted to $C := ENC(DAT, K_{P,sym})$ and only C is stored externally. To regain DAT again in clear the micro controller calls the decryption algorithm DEC adjoined to ENC in connection with the key $K_{P,sym}$.

In a similar way authenticity of DAT is protected by a message authentication code $MAC(DAT, K_{P,sym})$, whereas the message authentication code is stored together with the data in the external memory. For verification whether the data is unmodified the micro controller recalculates the output $MAC(DAT, K_{P,sym})$ and compares the calculated value with the stored one.

3.2 Secure Component Pairing

Up to now, we discussed a PUF based lightweight trust anchor mainly in the role of an alternative to a key storage in a security NVM. We will now consider an application scenario where such a token offers additional security benefit, even in the presence of security µC providing a secure NVM.

A popular architecture for secure embedded systems is often based on the approach to combine a first industrial micro controller µC1 with a security controller µC2, e.g., a smartcard IC or a TPM, to execute the necessary cryptographic algorithms and to hold the keys. If some data DAT generated by the first controller µC1 has to be protected cryptographically, µC1 sends DAT to µC2 for execution of the desired cryptographic operation using the appropriate key stored in the security NVM. This may be, e.g., the generation of a digital signature $sign(DAT)$ to DAT with the device specific private key k_{priv}. The result is transferred back from µC2 to µC1, and dependent on the application scenario, µC1 may send the cryptographically protected data in this example, the signed message [DAT,$sign(DAT)$], to an external recipient.

It is often assumed that this "*two-chip approach*" provides an overall solution to the security problems of embedded systems. However, this solution does not fit to attacker models in which the attacker has physical access to the embedded system and security IC. It is possible to manipulate the data stream from µC1 to µC2 on physically unprotected embedded devices. Even worse, it is possible to remove µC1 completely and to replace it by a tampered IC or to use µC2 on a different board. In any case, the security IC will execute the requested cryptographic operation without any additional access control.

Typically, access to a security IC is protected by knowledge and proof-of-possession of some kind of credential like a numerical pin or a password. Often an embedded system operates in an unattended way without a possibility to enter a pin or password, so that the credential again needs to be stored in a secure way on the insecure embedded system. To encounter these attack scenarios a fixed pairing between µC1 and µC2 is needed and, hence, the establishment of a secure channel connecting both. This fixed and secure pairing can be achieved with our universal key provision token at the beginning of the device life-cycle and could be realized in the following way:

- The device is inserted into a secure environment.
- The PUF-KT in µC1 is activated and the resulting symmetric key $K_{P,sym}$ is transferred via the device bus to µC2. The key $K_{P,sym}$ is permanently stored in the security NVM of µC2 .
- The device is removed from the secure environment and ready for operational use.

Fig. 2 depicts the just described pairing.

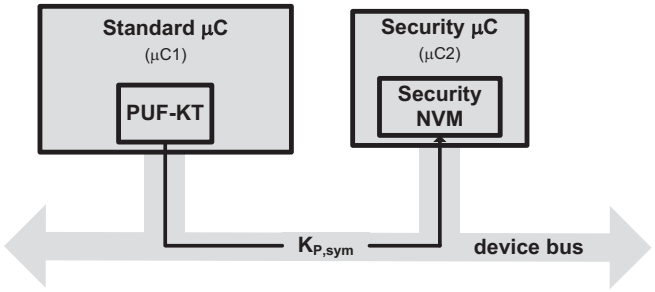


Fig. 2: Secure pairing Standard µC - Security µC

In operational service of the embedded device, at any power-on, PUF-KT in µC1 is automatically activated and produces again the key $K_{P,sym}$. Using a challenge-response protocol based on a symmetric authentication algorithm $AUTH$ the security controller µC2 verifies whether or not µC1 indeed possesses $K_{P,sym}$:

1) µC2 generates a random challenge C, computes $R = AUTH(C, K_{P,sym})$ and sends C to µC1.
2) On receipt of C the standard controller µC2 computes $R' = AUTH(C, K_{P,sym})$ a sends R' back to µC2.
3) If the received response R' coincides with R the security IC µC2 accepts µC1 as authentic.

With the acceptance of the received response by µC2 the secure pairing of the two controllers is established for the actual session.

In the sequel, additional keys might be derived to support encryption and data authenticity on the device internal link connecting both components. It is also possible to extend the described unilateral authentication to mutual authentication.

4. Criss-Cross PUF

A PUF's responses should be random and independent, but from an efficiency standpoint, as many response bits as possible should be extractable from a small area footprint. These two goals are conflicting, and depending on which of them is given more weight, different PUF constructions are preferrable: PUFs with a response space that scales linearly with their area (also called "weak" PUFs) are usually better at providing independent responses, provided that circuit design and operation do not introduce systematic effects [10], [11], since an independent circuit element is the dedicated source of each response bit. The most well-known representatives of this class are ring oscillator and SRAM PUFs [4]. On the other hand, so-called "strong" PUFs, such as the Arbiter PUF and its derivatives, offer a challenge-response space which scales exponentially with their area, but may be susceptible to machine learning attacks if their true entropy content is much smaller than the theoretical response space [12].

For dynamically generating different sets of keys $K_{M,priv}$ and $K_{M,sym}$, which can be changed over the lifetime of a device, a PUF with a large response space is needed. In addition, as the entropy per bit of a PUF is less compared to a physical random generator, more bits and a larger response space are necessary to achieve the same security level. To maintain its advantage over secured memory areas as key store, the manufacturing costs of the PUF should be minimal, meaning no additional process steps and minimal area. While ring oscillator and SRAM based PUFs fit the first criterion, the flexibility to repeatedly generate different sets of keys makes a strong PUF preferable.

4.1 Analysis of bistable ring based PUFs

The Bistable Ring PUF (BR PUF) [13] was a promising delay-based strong PUF with seemingly complex behavior, but it, as well as the improved derivative TBR PUF, has been attacked successfully with machine learning techniques [14], [15]. Based on the following discussion of the behavior of these PUFs on an analog level, we now will highlight the effects leading to this vulnerability, and subsequently propose the Criss-Cross PUF (CC PUF) as a new PUF primitive which in our view can overcome these problems.

The basic principle of the BR PUF consists of an even number of inverters connected in a ring, resulting in a

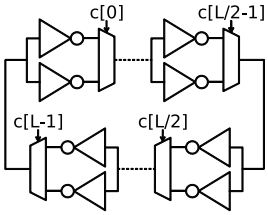


Fig. 3: Simplified BR PUF (reset mechanism not shown) of length L.

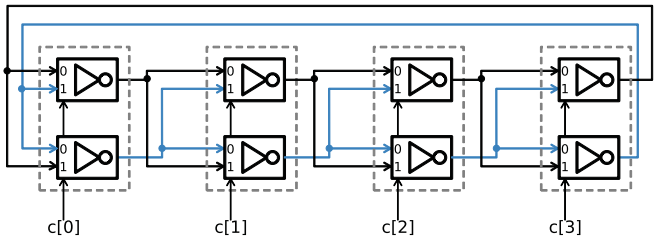


Fig. 4: Schematic representation of a four-stage Criss-Cross PUF.

system with two stable states. Via multiplexers controlled by challenge bits $c[i], 0 \leq i < L$, one of two possible signal paths per stage is selected, as indicated in Fig. 3. The circuit is made resettable by forcing it into a defined unstable state where all nodes are at the same value, e.g. by replacing the inverters by NOR gates with one input connected to a global reset signal. After leaving the reset state, the entire ring starts to oscillate in a traveling wave pattern as the inverters try to reconcile their input-output inconsistencies. But each stage has a slightly different driving strength and output load (e.g. routing capacitance), resulting in a different frequency response. The oscillation at different nodes thus gets out of phase, allowing inverters to achieve consistent input-output states one by one. When all inconsistencies in the ring are resolved, the ring settles into its stable state, which is determined by the interplay of the timing characteristics of the ring elements.

Looking at the schematic in Fig. 3, it is clear that a challenge bit $c[i]$ typically has very little influence on the timing behavior, as only one element of the ring is exchanged. If there is an element in the ring whose timing diverges strongly from the mean, it will dominate the entire ring's behavior, regardless of the other challenge bits. Even worse, the diverging timing may arise from an element which is not affected by the challenge, such as inter-stage routing capacitance, which is often neglected in simulations, but can be relatively large, especially in FPGAs. This can result in a settling state completely independent of the challenge. In practice, it is the position of the slowest element in the ring which has the highest influence on the final state, as arriving fast oscillations get damped there and the influence of other elements can be smoothed over.

This problem was correctly identified by the creators of the TBR PUF, in which each challenge bit swaps the position of two correspoding inverter elements in the ring [14]. However, the interconnection segments between stages remain unaffected by the challenge.

4.2 Benefits of the Criss-Cross PUF

Based on the above analysis, we propose the Criss-Cross PUF (CC PUF) as a new PUF primitive which overcomes this limitation. As shown in Fig. 4, the CC PUF consists of an even number of stages, each stage having two input signals and two inverters, and a corresponding challenge bit controls which input is applied to which inverter. The two inverter outputs are applied to the next stage, and so the two signals can "criss-cross" between the upper and lower tracks, with the challenge determining the order of the inverters as well as the interconnection segments traversed. Responses are generated as in the (T)BR PUF from the two possible settling states of a bistable ring after a reset signal (not shown in Fig. 4) is briefly applied to enforce an initial unstable state.

Defining that a '1' bit in the challenge means that the tracks are crossed, an interesting property of the CC PUF becomes apparent: If the number of '1' bits in the challenge (equivalently: its Hamming weight) is even, two independent bistable rings with length equal to the number of stages result, from whose settling states two theoretically independent response bits can be extracted. If the Hamming weight is odd, a single bistable ring with twice the length results, yielding one bit of settling state information, but generated from a more complex (and longer) stabilization process. In both cases, all ring elements remain active in determining the PUF's response, like in the TBR PUF. The CC PUF stage however requires only two input-side multiplexers and is thus easier to implement with balanced routing than the TBR stage. In FPGA implementations, it fits in two 3-input LUTs, and is also more area efficient as an ASIC implementation. The key difference however is that in the CC PUF, not only the traversal order of the inverter elements but also the order of the interconnection segments between stages is permuted by the challenge, which improves the behavior in the presence of significant and variable routing capacitance.

This is confirmed by transistor-level SPICE simulations, from which data is presented in Fig. 5. The histograms show inter-device Hamming distances from 100 Monte Carlo runs of TBR (top) and CC PUFs (bottom) under various conditions of the inter-stage capacitance: Without capacitance (left), and with a nominal capacitance of 100 fF whose random variability was set to 0, 10%, and 20%. The results show that the TBR is affected negatively by increasing influence of inter-stage capacitance, with its responses becoming more and more biased. Simulated BR PUFs (data not shown) were affected even more. The reverse seems to be the case

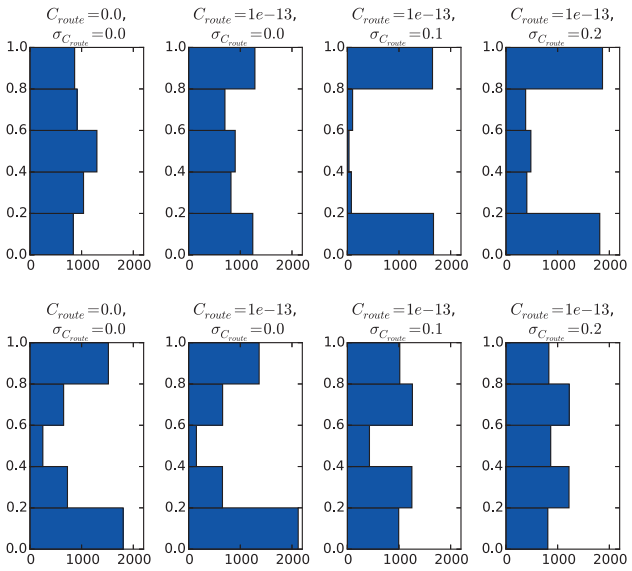


Fig. 5: Distributions of inter-device response Hamming distances for 100 simulated TBR (top) and CC PUFs (bottom) of length 4, with variation of the inter-stage capacitance C_{route} and its normalized standard deviation $\sigma_{C_{route}}$.

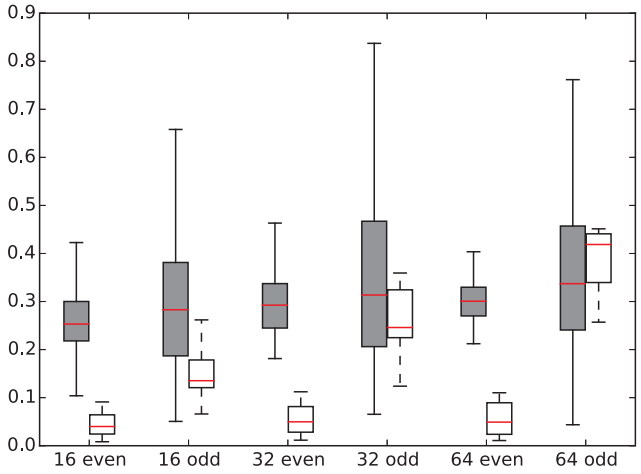


Fig. 6: Solid boxes and whisker lines show distributions of inter-device Hamming distances for CC PUF FPGA implementations of various lengths (pairwise comparisons between same subinstances on ten boards). White boxes with dashed whisker lines show intra-device distance between runs at 19 ambient temperature steps between -40 °C and 100 °C for the same implementations and challenge sets.

for the CC PUF, where the responses are in fact biased in the ideal case, but become more evenly distributed with higher influence of the inter-stage capacitance. This confirms our notion that the CC PUF can make use of inter-stage routing variability to improve response entropy.

4.3 FPGA Evaluation of the Criss-Cross PUF

A first evaluation of the CC PUF based on FPGA implementations was also performed with ten Xilinx Zynq Z-7020 SoC devices. Implementations of length 16 were placed four times on the chip, length 32 twice and length 64 once. In the following, different placements on the same chip are referred to as subinstances. Sets of 4096 challenges chosen randomly from each challenge space were split into "even" and "odd" sets according to their Hamming weights to obtain a good coverage of the challenge space while accounting for the different behaviour of the CC PUF depending on the challenge type. Although theoretically two response bits could be extracted per "even" challenge, only one will be used in the following for practical reasons.

Fig. 6 shows the obtained inter-device Hamming distances. The distributions for "even" challenges are more compact than those for "odd" challenges. The latter generated a relatively high rate of response timeouts (defined as the ring not stabilizing within 327.68 µs or 2^{15} clock cycles): On average, timeouts occurred with 8%, 15%, and 22% of "odd" challenges for 16, 32, and 64 stages, respectively, and virtually never with "even" challenges. The response was considered invalid in this case. The correlation between timeout rate and ring length is expected, as longer distances between any remaining inconsistent states traveling around the ring translate into a longer time until they can meet and neutralize each other, but it is not yet clear why the two types of challenges behave so differently in this regard. In summary, the inter-device distributions are tighter than in the simulation results, indicating that inter-stage influences have higher influence and variability than accounted for in the simulations. The median value of the inter-device distance is below 0.5 in all cases, however. This suggests that some systematic component remains in the response behaviour of all implemented PUFs.

As PUF responses need to be stable under varying environmental conditions, the CC PUF implementations were also tested in a temperature controlled chamber between -40 °C and 100 °C. In Fig. 6, the intra-device distances between all temperature pairs from this experiment are plotted next to the corresponding inter-device distances. For the responses from the "even" challenge sets, there is a safe margin to reliably reconstruct a nominal response through error correction and identify devices under large temperature variations. To give an accurate impression of the PUF's reliability, for the calculation of the intra-distance, any comparison involving an invalid response bit arising from a timeout was evaluated as different. Therefore, the intra-distances for the "odd" challenge sets are higher. If invalid responses are ignored, the intra-distance is in fact on the same level with both types of challenges.

Systematic influences could not be eliminated completely

in the FPGA implementations, resulting in some response bias and an average Shannon entropy of 0.6 bit per response bit. Two PUF response bits per key bit could thus be sufficient for key generation with the current implementation, but a much higher factor is both advisable for security and unproblematic due to the large challenge space.

5. Conclusion

PUFs offer an approach as lightweight trust anchors that provide cryptographic keys in a secure way and without a need to store them on external non-volatile memory. Additionally, the PUF based key can be applied for requirements like application specific key derivation, memory and firmware encryption or pairing of security ICs with a specific instance of a main controller.

As basis for these PUF applications, we implemented the Criss-Cross PUF, which improves BR PUF limitations regarding uniformity of response bits. First measurements of the FPGA implementation showed reduced bias values compared to the BRPUF, but does not reach the ideal value of 0.5 due to non-controllable side effects of the FPGA. Therefore, we will implement the Criss-Cross PUF as ASIC in a second step and expect further improved bias values. Finally, the Criss-Cross PUF, the PUF key provision token and some of the outlined application scenarios will be evaluated in a demonstrator of an industrial scenario.

Acknowledgment

This work was partially supported by the German Ministry for Education and Research (BMBF) project SIBASE under grant number 01IS13020.

References

[1] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Secure lightweight entity authentication with strong PUFs: Mission impossible?" in *Cryptographic Hardware and Embedded Systems - CHES 2014*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds. Springer Berlin Heidelberg, 2014, vol. 8731, pp. 451–475. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_25

[2] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," Cryptology ePrint Archive, Report 2014/977, 2014, http://eprint.iacr.org/.

[3] Y. Dodis, B. Kanukurthi, J. Katz, L. Reyzin, and A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 6207–6222, 2012. [Online]. Available: http://dx.doi.org/10.1109/TIT.2012.2200290

[4] R. Maes, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, University Leuven, 2012.

[5] A. Van Herrewege, "Lightweight PUF-based key and random number generation," Ph.D. dissertation, University Leuven, 2015.

[6] C. Bösch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on fpgas," in *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, 2008, pp. 181–197. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85053-3_12

[7] R. Maes, V. van der Leest, E. van der Sluis, and F. Willems, "Secure key generation from biased PUFs," in *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, 2015, pp. 517–534. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-48324-4_26

[8] M. M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/MDT.2010.25

[9] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography, Second Edition*, 2nd ed. Chapman & Hall/CRC, 2008.

[10] M. Pehl, A. R. Punnakkal, M. Hiller, and H. Graeb, "Advanced performance metrics for Physical Unclonable Functions," in *2014 14th International Symposium on Integrated Circuits (ISIC)*, Dec. 2014, pp. 136–139.

[11] F. Wilde, M. Hiller, and M. Pehl, "Statistic-based security analysis of ring oscillator PUFs," in *2014 14th International Symposium on Integrated Circuits (ISIC)*, Dec. 2014, pp. 148–151.

[12] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM conference on Computer and communications security - CCS '10*, Chicago, Illinois, USA, 2010, p. 237. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866335

[13] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The Bistable Ring PUF: A new architecture for strong Physical Unclonable Functions," in *2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, June 2011, pp. 134–141.

[14] D. Schuster and R. Hesselbarth, "Evaluation of Bistable Ring PUFs Using Single Layer Neural Networks," in *Trust and Trustworthy Computing*, ser. Lecture Notes in Computer Science, T. Holz and S. Ioannidis, Eds. Springer International Publishing, Jan. 2014, no. 8564, pp. 101–109. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-08593-7_7

[15] X. Xu, U. Rührmair, D. E. Holcomb, and W. Burleson, "Security Evaluation and Enhancement of Bistable Ring PUFs, Tech. Rep. 443, 2015. [Online]. Available: https://eprint.iacr.org/2015/443

A Dynamic Area-Efficient Technique to Enhance the Security of ROPUFs against Modeling Attacks

Fathi Amsaad, Chayanika Roy Chaudhuri, Atul Prasad Deb Nath, and Mohammed Niamat
Electrical Engineering and Computer Science Department, University of Toledo, Toledo, OH, USA
Email: {fathi.amsaad, croychaudhuri, aprasad }@rockets.utoledo.edu, mniamat@unet.utoledo.edu

Abstract - *Physical Unclonable Function PUFs are suitable security solutions for silicon technology chips including ASIC and FPGA chips. Despite the prevalence of numerous techniques for fabrications of Silicon PUFs (SPUFs), to the best of our knowledge, a well-established dynamic technique that can provide updated secret keys to improve ROPUF security against molding attacks does not exist. For this reason, an area-efficient technique that exploits an appropriate reconfiguration mechanism and dedicated FPGA resources to build a dynamic multi-stage ROPUF (d-ROPUF) structures is introduced. To determine the correlation between each structure and its performance, the normality of the generated RO frequencies is studied. Experiment results show that, a structure with fewer stages has higher performance compared to the one with more stages. To further validate the proposed technique, parameters of ROPUF loop Model are studied at normal operating conditions and compared to ROPUF parameters of earlier designs.*

Keywords: FPGAs, ROPUF Performance, Molding Attacks.

1. Introduction

Ring Oscillator PUFs are one of the most appropriate techniques for the security of silicon chips since it is a proven technique to provide high performance in terms of reliability and uniqueness of its generated response. However, compared to other silicon PUFs (SPUFs), ROPUF can only offer a limited number of challenge-response pairs (CRPs) for a generation of a secure binary responses. For this reason, ROPUF is categorized as weak SPUF which can be more venerable to molding attacks [1]. Number of CRPs in ROPUF design is linearly related to the number of components that are used to construct a ROPUF design whose behavior depends on the random manufacturing process variations. Thus, to overcome CRPs' limitation, more design components has to be incorporated in ROPUF design. Although, there are other constitutions of ROUF structure using different techniques to integrate more components into ROPUF (i.e configurable ROPUF), the generation of ROPUF secret keys are mainly based a single (static) behavioral of CRPs which can only extract a non-update secrete keys [2, 3]. Due to these limitations, an adversary may try all challenges and know the corresponding responses within a certain time that is linear to number of applied challenges [1]. However, a dynamic technique that offers multi-stages ROPUF can be highly unpredictable and less vulnerable to modeling attacks that aims to clone its structure. In this regard, a dynamic technique namely d-ROPUF that offers multiple CRPs behaviors to increases ROPUF unpredictability which in turn enhances its unclonablity against modeling attacks is proposed. The proposed technique is an area efficient ROPUF design that utilizes the dedicated FPGA logic (dedicated multiplexers, LUTs, fixed routings) to accommodate four multi-stages structures in a single CLB using Programmable XOR gates (PXORs) that control the individual RO frequencies. In addition, a reconfiguration mechanism is appropriately designed to automatically alter the behavior of CRPs and reconfigure the design to new structures with different stages (invertors). Hence, the design can generate updated secret keys and consequently becomes highly unpredictable and secure against modeling attacks.

Firstly, a detailed explanation on the significance of the proposed dynamic multistage technique is presented and then it is differentiated from prior ROPUF implementations (static CRPs techniques).

Secondly, we show how data samples are collected using the proposed techniques from the whole area of 30 Spartan-3E FPGA chips. To quantify the performance of each d-ROPUF structure, the normality of the generated sample RO frequencies and RO loop parameters are studied. In order to do that, RO sample frequencies of each d-ROPUF structure form 30 FPGAs is initially analyzed using two ROPUF normality parameters namely skewness and kurtosis. In addition, Kolmogorov-Smirnova (K-S) and Shapiro-Wilk (S-W) statistical tests are performed to determine the performance of d-ROPUF structures. ANOVA test is also used to compare the mean values of the average samples frequencies of each d-ROPUF structures. This comparison is necessary to ensure that the generated RO frequencies are represented by various data samples which reveal the differences in the CRPs behavior of the generated RO frequencies. The result shows that our RO sample frequencies are normally distributed with different CRPs behaviour.

Confirming that normality of d-ROPUF structures is the first step toward determining the correlation between RO sample frequencies and their reliability to generate unique secret keys. By selecting d-ROPUF structure with an appropriate number of stages that can generate reliable response bits, the number of RO sample frequencies that can be used to generate unique secret keys at varying operating conditions is maximized[4]. A higher difference between RO sample frequencies will ensure a higher performance in terms of its reliability and uniqueness [5]. In statistics, diverseness and variability are used to indicate the difference between data samples by measuring how far they are from each other[6]. For example, having n ROs with frequencies that are very close to each other, their diverseness and variability factors should always be close to 0. However, higher diverseness and variability values specify that data samples vary from each other which are very important for a higher ROPUF performance. Pearson's correlation factors 'r' (-0.93) and (-0.95), indicates a very strong inverse correlation between number of stages in each d-ROPUF structure and the average diverseness, and average variability of their RO sample frequencies, respectively. This shows that d-ROPUF structures with less number of stages exhibits high performance in terms of reliability and uniqueness compared to structure with higher number of stages. To estimate the effectiveness of d-ROPUF, quality parameters that are defined by prior researchers including ROPUF loop parameters, diverseness, variability are explored.

The reset of this paper is organized as follows: Section 2 covers research background. Section 3 explains the proposed design. Section 4 discusses the experimental results. Lastly, conclusions are drawn in Section 5.

2 Background

2.1 Basic and configurable Silicon PUFs

Arbiter PUFs (APUFs) [7], Ring Oscillators PUFs (ROPUFs) [8] are silicon based PUFs that take into consideration the process variations of Integrated Circuits (ICs) in order to produce random responses. In 2004, Lee et al [8] used a switch-box structure to create a race between two delay paths with an arbiter at the end. The basic circuit of APUF is presented in Fig. 1. Two identical delay paths are formed to produce a response bit based on the fastest path. The arbiter is placed at the end of the circuit (D-latch) to decide the winning signal that reaches first.

Due to their simplicity and high performance, ROPUF is one of the suitable security solutions for ASICs and FPGAs. Fig. 2 shows the original ROPUF design for extracting unique signatures. These signatures are produced using challenge-response mechanism and are difficult to clone or predict [7]. As shown in Fig. 2, ROPUF circuit compares the two frequencies fi and fj, in order to produce "0" or "1" output based on which frequency is higher. The measured process variations should be unique for each RO [7].

In 2007, a 1-out-of-k technique was also proposed by Devadas to improve the reliability factor of RO PUFs [7]. This technique was named the redundancy approach which selects 1-RO pair among k-RO pairs which has the maximum frequency difference. The major disadvantage of this technique lies in its area inefficiency since k times more area is wasted when it is implemented on FPGAs. In 2009, Maiti [6] introduced the notion of configurable ROPUFs (c-ROPUF) for better reliability. As shown in Fig.3 (a), he overcame the 1-out-of-k scheme drawback by offering a configurable design which occupies the same amount of area with more number of ROs. However, Maiti did not propose his method for stronger secret key production.

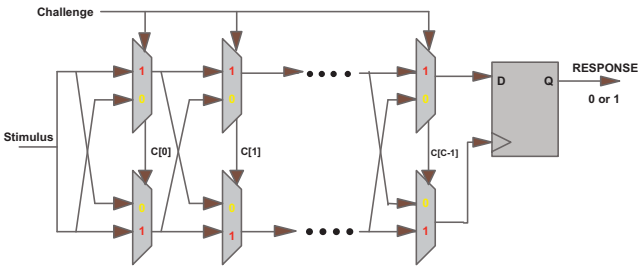


Fig. 1 An arbiter PUF design.

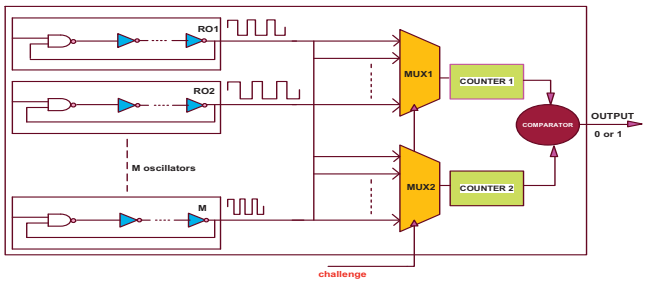


Fig. 2 Basic Ring Oscillator based PUF circuit.

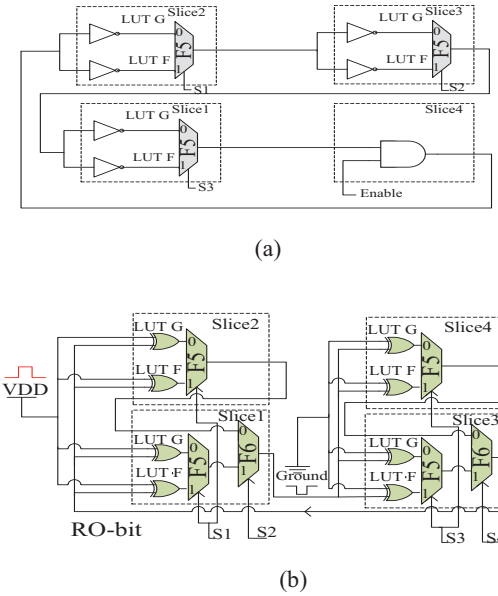


Fig. 3 c-ROPUF in one CLB: (a) Mait's design; (b) Amsaad's design.

Besides, the performance of Maiti's configurable design was not fully validated. An improved version of the configurable ROPUF by Maiti [5] that can generate stronger response bits while using the same amount of area is proposed by Xin [2] , and Amsaad in Fig. 3 [9].

2.2 Reconfigurable Silicon PUFs (rPUFs)

In 2005, Lim was the first one to propose the idea of reconfigurable PUF (rPUF) for an Arbiter PUF design [10]. His design was based on floating gate transistors for an Arbiter PUF. However, Lim did not clearly state how his proposed structure can be reconfigured, i.e. how his technique would specifically change the CRPs' behavior.

In 2009, Majzoobi presented a technique for implementing reconfigurable Arbiter PUF [11]. But, his implementation was based on a static PUF that does not satisfy the basic definition and conditions of a secure rPUF.

In 2009, Kursawe first presented the concept of reconfigurable optical PUF [3] as a physical structure with light scattering particles. Kursawe defined reconfigurable PUF (rPUF) as a PUF that is equipped with an appropriate mechanism to automatically convert its structure into a new structure with a new unpredictable challenge-response behavior [3]. In order to achieve this, the reconfiguration mechanism should be separately designed and controlled. Thus, the mechanism can alter the behavior of CRPs without the applied challenge affect. Kursawe proposed to reconfigure his structure with the help of physically state reposition of scattering particles method (polarization). Polarization is defined as the internal structure of the design when it's exposed to a laser beam outside the normal operating conditions [3]. Kursawe laid out a theoretical example of optical rPUF and its mathematical model but practical aspects and analysis were not specified. As seen in the previous figures for the static SPUF designs (APUF, ROPUF and c-ROPUF), the challenge bits are either totally or

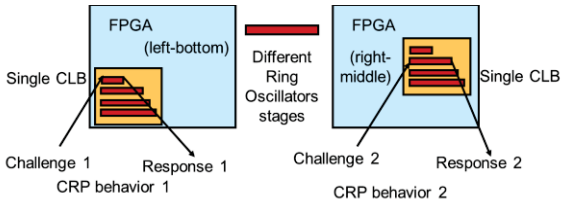


Fig. 4. Proposed d-ROPUF general scheme.

partially used to configure their structures which makes the configuration mechanism possibly reversible. Based on the definition of reconfigurable PUFs (rPUF) by Kursawe, neither original SPUFs nor the configurable SPUFs can be considered as reconfigurable challenge response behaviour (rPUFs) [2]. Both basic and c-ROPUFs are FPGA friendly techniques. However, c-ROPUF provides more responses bits for better ROPUF security.

As far as our knowledge goes, prior research is focused on the static ROPUF (basic and configurable ROPUFs) which considers the implementations and performance aspects of static structure. The drawback of static ROPUFs is that, when the design is configured with high number of ROs, due to implementation of the design using a single structure that has a fixed number of stages; the design can only generate RO sample frequencies within a certain range. On the other hand, dynamic ROPUFs can always generate RO sample frequencies within different range of frequencies due to the design's ability to reconfigure itself into a new structure with different number of stages. This also increases CRPs space and updates their behavior which makes it harder for the adversary to try all possible challenges and uncover the corresponding responses (clone or model). Once the behavior of CRPs is known to an attacker, he can easily clone the entire ROPUF structures and store them into fake chips using memory device in order to hack in to a certain system. Having multi-stages ROPUF in the same CLB, whose behaviors are automatically altered, not only improves the efficiency of ROPUF in terms of the occupied area, but also enhances its unpredictability by enabling the generation of more complex cryptographic keys. As seen in Fig. 4, using dynamic ROPUFs updated secret keys can be randomly extracted from different structures implemented on different FPGA areas with the help of new CRPs behaviours. As a result, the extracted cryptographic keys become more complex and highly unclonable. In the next section we illustrate the details of implementation and analysis of the proposed technique namely dynamic ROPUF (d-ROPUF) and show how it can be useful to improve ROPUF security against modelling attacks.

2. Implementations of the Proposed Technique

In this section we explain how the design is implemented and data is collected from 30 Spartan-3E/100k FPGAs under normal conditions. As shown in Fig. 5, for area efficiency the proposed design is mapped inside one CLB taking advantage of the identical nature of internal routings that connect LUTs to two levels of dedicated MUXs (F5, F6) with fixed routing delays [9]. Internal routings guarantees identical RO loops in different CLBs and elements the percentage of noise caused by dynamic routings [9].A ring oscillator loop in a d-ROPUF structure is a circuit composed of an odd number of inverters that are serially connected. As shown in Fig. 5, single CLB in Spartan-3E FPGAs contains four slices and eight LUTs (2 LUTs per slice). Each of these eight LUTs can be configured as an inverter or a buffer. In order to have odd number of inverters (1, 3, 5, and 7), seven LUTs are configured as inverters and one LUT as a buffer using PXOR gates. These LUTs are connected using internal CLB routings and four different structures are built inside a single CLB. As seen in Fig. 5, the red lines connect the enable signal (VDD) to activate the selected CLB and also configure LUTs as controlled inverters (PXORs). There is only one LUT that is configured as a buffer (connected to ground signal) which adds extra delay to keep the generated frequency for one-stage inventor RO structure less than 300 MHz, that is the maximum operational value of Spartan 3E FPGA frequency. Likewise, different LUTs are shown in different colors and connected using different colored lines to show how four ROPUF structures are mapped using internal CLB routings. For example, one stage structure is only implemented using the green LUTs, the three stages is implemented using green and blue LUTs. The five stages is implemented using green, blue and purple LUTs while the seven stages is implemented using green, blue, purple and brown LUTs. Fig. 6 shows the proposed technique for an FPGA area. A 0.1 ms delay is provided before the activations of each counter, so that the signal is stabilized before the actual RO frequency measurement starts. To implement the design in the entire area of FPGA, we divided each chip to top and bottom area (120 CLBs).

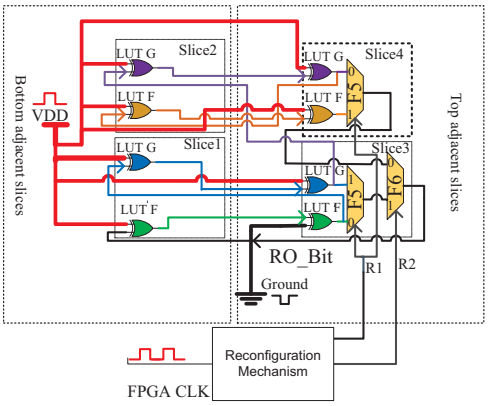


Fig. 5. Proposed d-ROPUF in a single CLB of Spartan 3E FPGA.

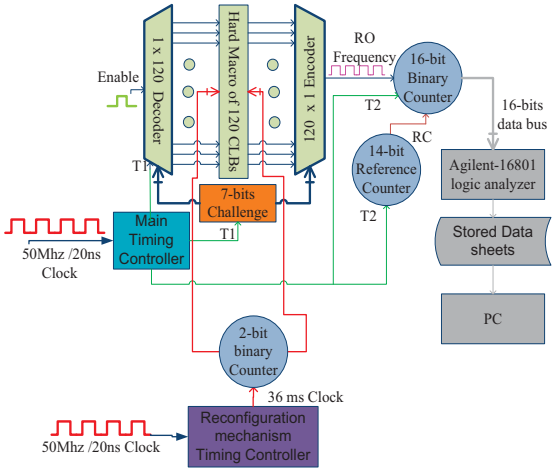


Fig.6 Proposed technique for an FPGA area (120 CLBs).

To avoid self-heating noise of the neighboring ROs, each RO is activated in our design for a shorter time 0.1 ms (activation period) with a step size of 5,000 clock cycles [13].A deactivation period of 0.1 ms is also allowed before activating the next RO. The timing controller shown in Fig. 6 controls the activation, deactivation periods of the individual ROs, the challenge generator, 120 decoder & encoder, the main and reference counters using T1 and T2 signals. Once the challenge generator and the T1 signal, the input challenge is produce to the decoder, encoder and a single RO is activated for 0.1 ms. At the end of this period, the main counters receive T2 from the timing controller and starts counting the number of clock cycles that is generated by the activated RO. At the same time, the reference counter receives the T2 signal and counts for 0.1 ms (5,000 clock cycles) before it stops the main counter by sending RC signal. At the end of the second 0.1 ms period, the main reference and main counters will be deactivated and a control RC signal is sent by the reference counter to the main counter. Once the main counter receives RC signal, it stops counting and forwards the counted number of clock cycles through a 16-bit data bus to an Agilent logic analyzer. A third 0.1 ms is given before the main counter sends T1 signal to activate another RO.

As a result, 120 Sample frequencies for each RO structures are stored for each d-ROPUF structure in form data sheets in the logic analyzer before it is collected and analyzed offline. Individual ROs for each d-ROPUF structure are activated from the top and moves down to the bottom of each column of the CLBs. Average frequency of 10 data samples for the entire individual ROs is considered for analysis. This requires a total time for the 10 runs per all d-ROPUF structures, which is 10 × 144 = 1.44 s.

Fig. 7 shows the actual implementations of the proposed technique mapped in bottom FPGA areas. As shown in the figure, the reconfiguration mechanism is placed separately in the top left area of FPGA (12 CLBs), while the other associated RO logic (decoder, encoder, challenge generator, ets) are built in the top right area. The bottom area of FPGA (120 CLBs) is completely occupied by the instantiated d-ROPUF structures in 120 CLBs.

A synchronous binary counter counts from 0 to 2N-1, where N is the number of bits (flip-flops) in the counter. As seen in Fig. 8, the generation of R1 and R2 singles is separately controlled. The reconfiguration mechanism is properly designed a part of the input challenge using 21-bit up counter that uses 2ns clock cycle as an input clock and produces a 36 ms to a 2-bit up counter. The dedicated Muxes in each CLB (F5, F6) in each CLB selects different RO structures using the received signals from the reconfiguration lines R1 and R2. Thus, the behavior of challenge-response of each d-ROPUF structure is altered on the account of their number of RO stages and provides an updated response bits.

To implement the reconfiguration mechanism a one flip flops per each 1-bit is needed. Since Spartan 3E FPGA has four dedicated flip flops in each CLB, a proper placement constraint has been applied using VHDL to separately control the implantation of the reconfiguration mechanism within 12 CLBs (23 flip flops) as seen in Fig. 7. For the seek of simplicity, to collect data from the entire d-ROPUF structure (120 RO frequencies), R1 and R2 is incremented by 1 using the 2-bit up counter (R1R2= 00, 01, 10 or 11) every 0.3 ms × 120 CBLs= 36 ms period.

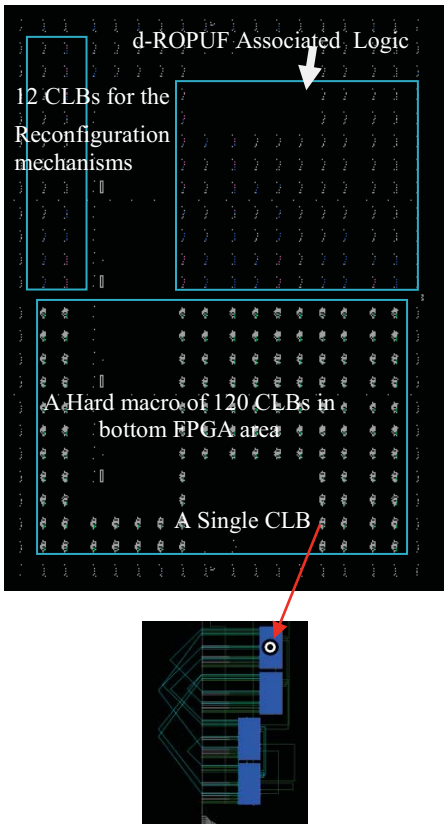


Fig. 7 Proposed d-ROPUF technique in the bottom FPGA.

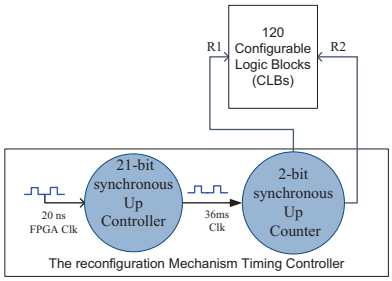


Fig. 8 The design of the reconfiguration mechanism.

4. Discussion of Experimental Result

4.1 The normality of RO sample frequencies

Data samples are collected and analyzed for each d-ROPUF structure (120 ROs in each FPGA region). To collect data samples, the number of clock cycles of an active RO is measured using Agilent-16801 logic analyzer that can correctly recognize the non-uniform pattern of the individual RO frequencies. The oth sample RO frequency $f_{l,m,n,o}$ $(1 \le o \le 10)$ of the n^{th} RO $(1 \le n \le 240)$ of the m^{th} FPGA area $(1 \le m \le 2)$ in the l^{th} FPGA $(1 \le l \le 5)$ is calculated using the following equation [2]:

$$f_{l,m,n,o} = (CC_{RO} \times 50\ MHz) \ / \ CC_{ref} \quad (1)$$

where, CC RO is the number of clock cycles of the active ROs and CC ref is the step size of FPGA clock cycle. After careful assessment, a step size of 5,000 cycles was selected for the reference clock cycles. The default FPGA internal clock (50 MHz) is used as a reference clock. To reduce the effect of local noise on RO frequencies, average frequency $f_{avg_{l,m,n}}$ of 10 data samples for each ring oscillator is calculated as follows [12]:

$$f_{avg_{l,m,n}} = \frac{1}{o}\sum_{i=1}^{o} f_{l,m,n,o} \quad (2)$$

$f_{avg_{l,m,n}}$ value in equation 2 is a good measure for the average sample frequency of the individual ROs (d_{AVG}) which is a fixed value for all ring oscillators resulting from architectural and technological parameters [4,12]. As shown in Fig. 9, data samples are collected from the entire area of 30 Spartan-3E FPGAs (240 ROs) using 16801-A Agilent logic analyzer. Average RO frequencies for a FPGA out of 30 FPGAs are calculated using equation as follows:

$$Avg\_\text{FPGA}_l = \frac{1}{m \times n}\left(\sum_{j=1}^{m}\sum_{k=1}^{n} f_{avg_{l,m,n}}\right) \quad (3)$$

The mean and median values are computed with the help of IBM-SPSS software as follows:

$$Mean = \frac{1}{l}\left(\sum_{i=1}^{l} Avg\_FPGA_l\right) \quad (4)$$

$$Median = \frac{1}{2}(N+1) \quad (5)$$

where n represents RO frequencies of 240 data samples for 30 FPGAs.

Table 1 shows that, the value of Mean and Median of all structures are very close to each other which indicates that RO sample frequencies are more likely normally distributed. As shown in Fig. 10, a distribution is considered normal (symmetric) when the right and left sides of the graph are approximately mirror images of each other. Kurtosis value often provides a measure of sharpness of the central distribution peak. A distribution that has a less sharp peak (more readings near tails) has negative Kurtosis (Kurtosis < 0). On the other hand, a distribution that has a sharper peak than normal (more readings near center) has positive Kurtosis (Kurtosis > 0). For a symmetric distribution, the Skewness and Kurtosis values are equal to zero. As seen in Table 2, one stage structure is skewed to the right since it has a positive Skewness value. Furthermore, one and seven stage structures are closest to the symmetric distribution because its Skewness is closest to zero. On the other hand, three and five stage structures appear to have the highest Skewness to the left. As far as Kurtosis is concerned, since three and five stages are closer to zero, they tend to have more readings near the center and are closer to normality when compared to one and seven stages which have more data near the tails (negative Kurtosis) and are far away from the center. Fig 5 (a), (b), (c) and (d) show the distribution of average sample frequencies of each structure for all 30 FPGAs. To accurately determine the normality of RO frequencies for each structure, we perform two different statistical tests using IBM-SPSS software namely: Kolmogorov-Smirnova and (K-S) Shapiro- Wilk (S-W) for each r-ROPUF structure.

Fig. 9 Experimental setup

Table 1 Performance Parameters of r-ROPUF structures

Statistical Measure	ROPUF Structures			
	One stage	Three stages	Five stages	Seven stages
RO Sample Frequencies (N)	240	240	240	240
Mean	264.88	252.34	148.82	133.58
Median	264.80	252.44	148.15	133.55

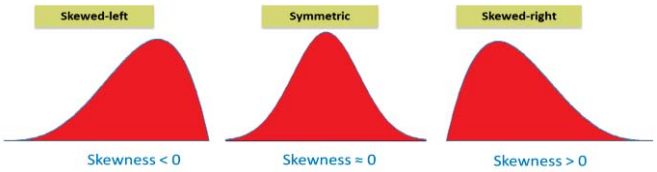
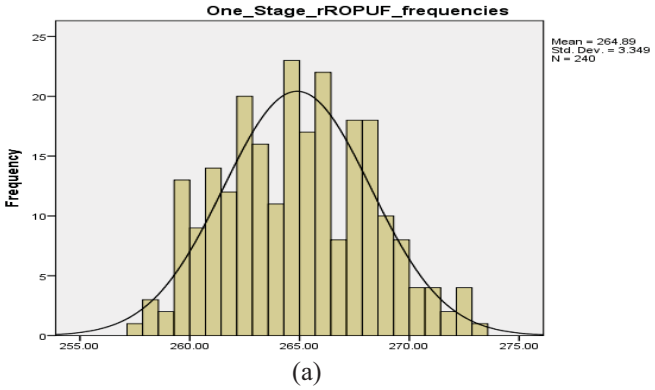


Fig. 10. The Skewness values of the distribution.

Table 2 Skewness and Kurtosis values of d-ROPUF structures

d-ROPUF Parameter	d-ROPUF structures			
	One Stage	Three Stages	Five Stages	Seven Stages
Sample RO Frequencies	240	240	240	240
Skewness	.129	-.146	-.164	-.129
Kurtosis	-.606	.494	-.210	-.350



(a)

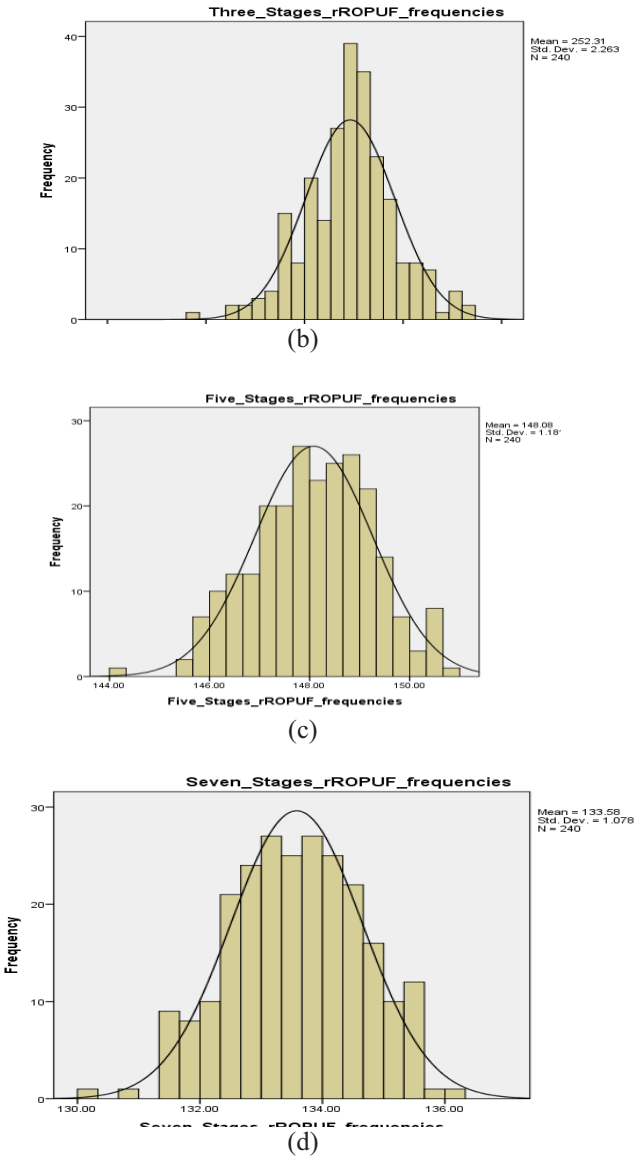


Fig. 11 The distribution of average RO frequencies for r-ROPUF structures: (a) one stage, (b) three stages, (c) five stages, and (d) seven stages.

For K-S test, the critical p-values=0.04 at a significant level ∝=5% are used to determine the normality with the following hypotheses taken into consideration:

H0: The data follow a normal distribution
Ha: The data does not follow the normal distribution

The critical region rejects H0 if the statistical value > 0.04. From Table 3 it is observed that, the five and seven stages d-ROPUF structures are at the lower bound of the significance value which is 0.200. However, the one and three stages reconfigurations are closer to zero (0.003 and 0.29) which indicates that their RO frequencies are more random.

Table 3 Tests of normality of the d-ROPUF

d-ROPUF Structures	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
One stage	.074	240	.003	.982	240	.004
Threes stage	.061	240	.029	.991	240	.142
Five stage	0.043	240	.200*	.992	240	.193
Seven stages	0.041	240	.200*	.994	240	.413
Critical value =0.04, *This is a lower bound of the true significance.						

This means that the null hypothesis is not rejected for all r-ROPUF structures, because the p-value for each one is <= 0.4, which consequently shows that RO frequencies of all r-ROPUF structures are normally distributed. As seen in table 3, Shapiro-Wilk test in the same table shows that at the 98% level we accept the normality of one stage d-ROPUF structures, and at the 99% level we accept the normality of other stages for the remaining structures. IBM-SPSS software is also used to calculate the one-way analysis of variance (ANOVA test) to compare the mean values of frequencies of the d-ROPUF reconfigurations using F distribution. The software uses a significant level of α=0.5. The critical value, as found by the IBM-SPSS software, is p-value=0.41.

Since P < α, this indicates that the group means of d-ROPUF configurations are taken from the data samples with different frequencies. The use of ANOVA test is crucial here since it demonstrates that each r-ROPUF structure can produce different frequencies based on the unique behavior of its CRPs.

4.2 ROPUF loop Model

The following equation [2] is used to analyze d-ROPUF loop parameters:

$$d_{RO} = d_{AVG} + d_{PV} + d_{NOISE} \tag{6}$$

Average diverseness and variability that represents the standard deviation and variance values of RO frequencies generated from the entire area of 30 FPGAs (top and bottom areas) are calculated as follows:

$$Diverseness_l = \sqrt{\frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}(avg\_FPGA_l - f_{avg_{l,m,n}})^2} \tag{7}$$

$$Avg\_diverseness = \frac{1}{l}\left(\sum_{i=1}^{l} Diverseness_l\right) \tag{8}$$

$$Avg\_Variability = \left(Avg\_Diverseness\right)^2 \tag{9}$$

As seen in Table 4, average diverseness and variability of the individual RO frequencies decrease with increase in the number of stages. This in turn indicates lesser the number of stages, more reliable is the ROPUF response [4, 12]. d_{AVG} is fixed for ROs which represent the average frequency of the identically instantiated RO loops.

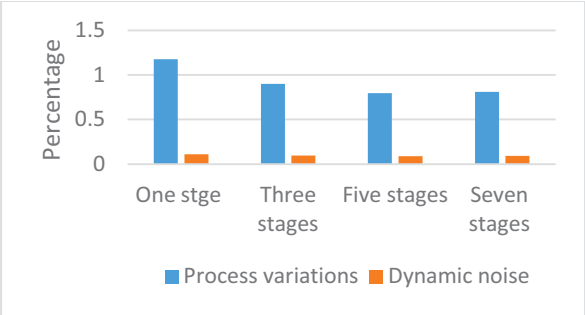


Fig. 12 Percentage of static process variation and dynamic noise variation

Table 4 Performance Parameters of d-ROPUF structures

Statistical Measure	ROPUF Structures			
	One stage	Three stages	Five stages	Seven stages
Avg. Diverseness	3.35	2.27	1.18	1.08
Avg. Variability	11.22	5.14	1.40	1.16
Rang	15.7	13.7	6.6	5.8

Table 5 Comparison between five stages dynamic and static ROPUF

Comparison	Maiti five stages ROPUF [5]	Proposed five stages d-ROPUF
Avg. RO Freq.	205.1MHz	148.82 MHz
Avg. process Var. (d_{pv})	0.75%	0.81%
Avg. Noise Var (d_{Noise}).	0.025%	0.0089%
(d_{noise}) / (d_{pv}) ratio	0.03	0.01

However, the percentage of intra-die process variation value d_{PV} varies from one RO loop to other which is shown in Table 4 and computed using the following ratio [2]:

$$d_{PV=}(Avg\_diverseness / Mean)\times 100\ \% \qquad (10)$$

Average noise variations (d_{NOISE}) for RO loops is calculated for each structure in equation (11) and shown in Fig. 6 as follows [2]:

$$d_{NOISE} = \sum_{i=1}^{l} \sqrt{(Diverseness_l / Mean)\times 100\%} \qquad (11)$$

As shown in Table 5 , average process variation (d_{PV}) is in the range 0.8% - 1.18%, with an average of 0.92% and a standard deviation of 0.18%. A higher process variation implies a higher d-ROPUF performance. [3, 8, 12]. From the same figure we notice that, a d-ROPUF structure with less number of stages has higher static process variation compared to the one with more number of stages. This indicates that, a d-ROPUF structure with less number of stages is more reliable to authenticate chips. Table 4 also depicts the distribution of average noise variation for ROPUF structures which is in the range 0.0089% - 0.012%, with an average of 0.0095% and a standard deviation of 0.0089%. It is also observed that, d-ROPUF structure with less number of stages has lower noise d_{NOISE} compared to d-ROPUF structures with more number of stages. Table 4 shows a comparison between five stages d-ROPUF loop parameters values and five stages ROPUF in [5]. Even though the average frequencies obtained in [5] is higher than our obtained average frequency, the ratio between the average static process variation and average dynamic noise variation is smaller for our five stages d-ROPUF structures. Thus, it is expected that our proposed design exhibits high performance under different conditions [2, 11].

5. CONCLUSION

In this paper, a dynamic design namely, dynamic ROPUF structures (d-ROPUF) with different challenge/response behaviour is proposed to enhance ROPUF security against hardware vulnerabilities and attacks. The design is implemented and evaluated on 30 Spartan 3E FPGA chips. Moreover, the normality and performance aspects of four different r-ROPUF structures are covered in details. It is shown that the obtained RO frequencies are normally distributed. Our result shows that the d-ROPUF exhibited fairly good performance in terms average diverseness, average variability at normal operating conditions. Using a newly proposed parameter, RO variability, we determined that the performance of r-ROPUF structures can be increased by having less number of stages and vice versa. In addition, high percentage of static to the dynamic variation ensures good PUF reliability in terms of intra-chip Hamming distance.

REFERENCES

[1] C. Herder, M. D. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *in Proc. of IEEE*, vol. 102, no. 8, pp. 1126-1141, Aug. 2014A.

[2] Xin Xin, Jens-Peter, and Kaps Kris, "A Configurable RO-Based PUF for Xilinx FPGAs," *Proc.* 14th Euromicro.

[3] K. Kursawe, "Reconfigurable physical unclonable functions – enabling technology for tamper-resistant storage," *HOST* , 2009.

[4] M. Mustapa, M. Niamat, M. Alam, and T. Killian, " Frequency uniqueness in ring oscillator Physical Unclonable Functions on FPGAs," in *the 56th MWSCA IEEE Proceeding,* Aug 2013.

[5] A. Maiti and P. Schaumont, "Improving the quality of physical unclonable function using configurable ring oscillators, *in Field Programmable Logic and Applications – FPL* 2009.

[6] Venkatarama Krishnan, "Statistics for the Behavioral Sciences," *ISBN-13: 978-1-111-83099-1*, 9th Edition.

[7] G.E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," *ACM Press*, 07.

[8] Lim D., Lee, J.W., Gassend, B., G.E., Van Dijk, M., Devadas, S., "Extracting secret keys from integrated circuits," *IEEE Transactions on (VLSI) Systems,* 2005.

[9] F. Amsaad, T. Hoque, M. Niamat "Analyzing the performance of a configurable ROPUF design controlled by programmable XOR gates," in Circuits and Systems (MWSCAS), 2015.

[10] D. Lim, "Extracting Secret Keys from Integrated Circuits," Master's thesis, MIT, May 2004.

[11] M. Majzoobi, and M. Potkonjak, "Techniques for Design and Implementation of Secure Reconfigurable PUFs," *ACM Transactions on Reconfigurable Technology and Systems*, 2009.

[12] A. Maiti, J. Casarona, and P. Schaumont, "A large scale Characterization of RO-PUF," *Proc. HOST,* 2010.

[13] P. Sedcole and P. Y. K. Cheung, "Within-die delay variability in 90nmFPGAs and beyond," In *IEEE FPT Proc.* 2006, pp. 97-104.

An Automated Tool for Evaluating Hardware Trojans and Detection Methods

Nicholas Houghton, Samer Moein, Fayez Gebali, and T. Aaron Gulliver
Department of Electrical and Computer Engineering
University of Victoria
P.O. Box 1700 STN CSC
Victoria, B.C. V8W 2Y2
Email: {nhoughto, samerm, fayez, agullive}@uvic.ca

***Abstract*—The growing concern for hardware security has spawned numerous trojan detection methods. Due to the complexity of integrated circuits (ICs), detection methods developed thus far have only been successful for specific trojans. This makes it difficult to compare hardware trojans and the performance of detection methods. In this paper, a systematic and automated approach to analyzing the characteristics of trojans and detection methods is presented. Universal adoption of the techniques in this system will aid in collaboration and standardization in the field. It will also provide a centralized database of existing hardware trojans and detection methods. A discussion of the automated tool design is given including a case study to demonstrate its usefulness.**

I. Introduction

The field of hardware security is relatively new and as a consequence lacks accepted techniques for analysis and comparison. The variety of Integrated Circuit (IC) design and construction techniques has resulted in a diversity of structures, behavior and locations of hardware trojans. As a consequence, the majority of detection methods have been designed for specific trojans. As an example, side-channel methods such as power [1]–[5], current [6], [7], delay [8]–[10], and Electromagnetic (EM) radiation [11], or a combination of these techniques [12], all rely on certain trojan attributes to achieve success. Each of these methods can fail with only a minor change to the trojan. In [13], the nonuniform nature of currently available methods was discussed, as well as the lack of a means of analyzing and comparing them.

Hardware trojans are comprised of attributes which can be used to characterize them, such as the entry point in the circuit, effect, and location. A taxonomy for hardware trojans which incorporates the IC life-cycle and characteristics was presented in [14]. This taxonomy can be employed to detect hardware trojans as well as analyze and compare detection methods. With an effective means of characterizing trojans, designers and attackers can select the best approach to detection given the available resources. However, analysis can be difficult and prone to error. Thus, a tool called the Hardware Trojan System (HTS) has been developed to automate this process. It allows users to quickly and easily select trojan characteristics based on their observations. These characteristics are input to the system and attribute tables are used for analysis. The HTS automates the necessary computations, provides centralized documentation, and compiles a database of trojans and detection methods. This database can be used by developers and attackers to search existing trojans and detection methods for design purposes.

The contributions of this paper are as follows.

1) A new technique is proposed for evaluating trojans and detection methods.
2) A web-based tool which automates the evaluation and comparison processes is presented.
3) A database to store the analysis results is developed.

The remainder of this paper is organized as follows. Section II discusses the new comprehensive taxonomy proposed in [14]. The attributes are studied and trojan risk (severity) and detection effectiveness (coverage) values are assigned. Sections III and IV provide a description of the on-line tool while Section V provides a case study to demonstrate its use. Finally, some concluding remarks are given in Section VI along with suggestions for future work.

II. Hardware Trojan Taxonomy

Several hardware trojan taxonomies have been proposed in the literature [15]–[18]. In [15], trojans were organized based solely on their activation mechanisms. A scheme based on the location, activation and action of a trojan was presented in [16], [17]. However, these approaches do not consider the manufacturing process. A taxonomy was proposed in [18] which employs five categories: insertion, abstraction, activation, effect and location. While this is more extensive than previous methods, it fails to account for the physical characteristics of the trojan. The taxonomy proposed in [14] is the most comprehensive as it considers all known attributes a trojan may possess and also their relationships. This makes it possible to not only analyze the merits of a detection method compared to others but also allows for the investigation of the coverage of a method in detecting trojans. This taxonomy is comprised of thirty-three attributes organized into eight categories as shown in Fig. 1. These categories can be arranged into four levels as shown in Fig. 2.

1) The **insertion** (chip life-cycle) level comprises the attributes pertaining to the IC production stages.
2) The **abstraction** level corresponds to where in the IC abstraction the trojan is introduced.
3) The **properties** level comprises the behavior and physical characteristics of the trojan. It contains the taxonomy

ISBN: 1-60132-445-6, CSREA Press ©

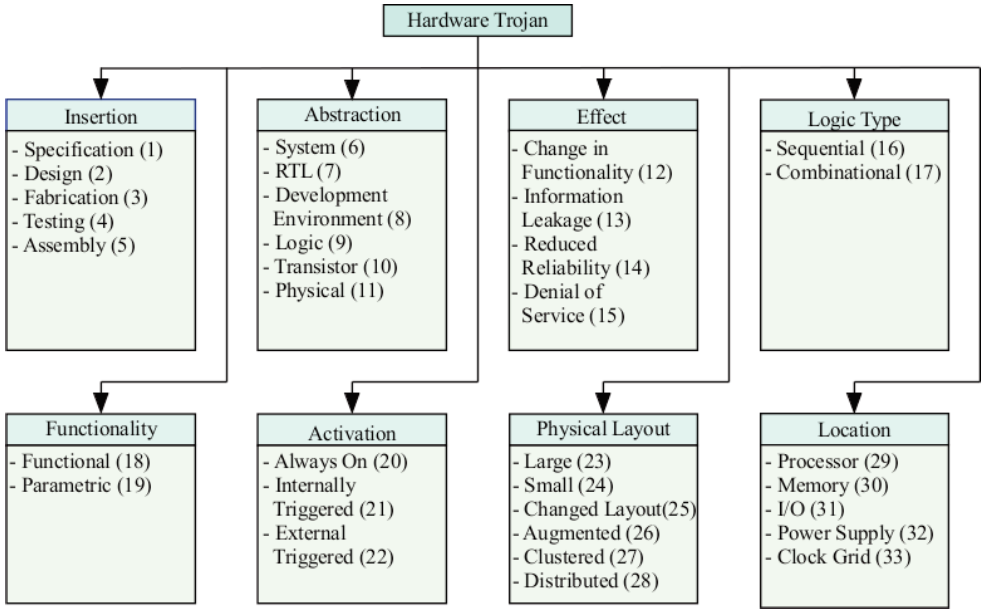


Fig. 1: The hardware trojan attribute taxonomy [14].

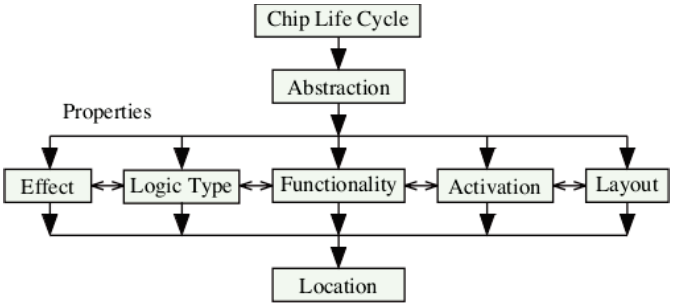


Fig. 2: Hardware trojan life-cycle levels [14].

k	Attribute	I_E	C_E
12	Change in Functionality	1	2
13	Information Leakage	2	4
14	Reduce Reliability	3	1
15	Denial of Service	4	2
12 & 13		5	6
12 & 14		6	3
12 & 15		7	4
13 & 14		8	5
13 & 15		9	6
14 & 15		A	3
12 & 13 & 14		B	7
12 & 13 & 15		C	8
12 & 14 & 15		D	5
13 & 14 & 15		E	7
12 &13 & 14 & 15		F	9

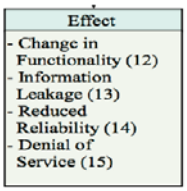


Fig. 3: Identification, coverage and severity values for the effect category [14].

categories *effect*, *logic type*, *functionality*, *activation* and *layout*.

4) The **location** level corresponds to the location of the trojan in the IC.

The properties level has the following categories.

- The **effect** category describes the disruption or effect a trojan has on the system.
- The **logic type** category describes the circuit logic that triggers the trojan, either combinational logic or sequential.
- The **functionality** category differentiates between trojans which are functional or parametric.
- The **activation** category differentiates between trojans which are always on or are triggered.
- The **layout** category is based on the physical characteristics of the trojan.

A hardware trojan or detection method possesses a number of the thirty-three attributes in Fig. 1. Each trojan or detection method is assigned two values for each category, and these are given in the vectors I_P and C_P. I_P is the identification vector which is used to differentiate between the possible attribute combinations. For a detection method, C_P is the effectiveness or coverage vector while for a trojan it is the severity or risk vector. Fig. 3 provides an example of the attribute combinations and the corresponding values for the effect category (denoted by I_E and C_E).

Table I provides a comparison of two hardware trojans. Trojan A has a lower severity than Trojan B in the insertion category, denoted by C_R. This indicates that Trojan B can be inserted in more stages of the manufacturing process than Trojan A. Table II gives a similar comparison between two detection methods. The method in [19] has a higher coverage in the effect category than the method in [20], indicating that it can detect more trojans. Table II and Fig. 3 indicate that the method in [19] is capable of detecting attributes 12, 13 and 14 whereas the method in [20] is only capable of detecting attribute 12. Any trojans or detection methods can be compared in a similar manner. Note that when a comparison is performed the techniques may have equal values in some categories. For example, from Tables I and II, the detection methods in [19]

TABLE I: Severity Vectors for Two Hardware Trojans

Technique	Parameters (I_P)								Severity (C_P)							
	I_R	I_A	I_E	I_L	I_F	I_C	I_P	I_O	C_R	C_A	C_E	C_L	C_F	C_C	C_P	C_O
Trojan A [14]	2	6	2	1	2	1	7	7	2	6	4	1	2	1	5	2
Trojan B [14]	3	3	1	2	1	2	8	1	3	3	2	2	1	3	6	1

TABLE II: Coverage Vectors for Two Hardware Trojan Detection Methods

Technique	Parameters (I_P)									Coverage (C_P)								
	I_R	I_A	I_E	I_L	I_F	I_C	I_P	I_O	I_G	C_R	C_A	C_E	C_L	C_F	C_C	C_P	C_O	C_G
[19]	3	3	B	1	2	4	7	V	1	3	3	7	1	2	3	5	5	2
[20]	3	3	1	2	1	4	7	V	4	3	3	2	3	1	3	5	5	3

and [20] as well as Trojan B all have the same value in the C_R category, which indicates that they have the same coverage and severity, respectively. Thus, the methods in [19] and [20] are capable of detecting Trojan B in the C_R category.

III. The Detection Tool

When a new method of detecting hardware trojans is developed, it should be evaluated using current trojans and compared with other detection methods. The HTS *detection tool* allows developers to investigate trojan detection methods systematically and perform a quantitative analysis. A user can select values for each of the eight categories as well as provide the method name and description. The corresponding vector is then stored in the database. The HTS provides a number of automated tools such as for the *trojan classification* in [21] which employs the categorization scheme in [14]. This tool provides a means of determining the severity of trojans, and this can be saved in the database for use with the detection tool. The detection tool allows users to search the database for detection method coverage (effectiveness) and trojan severity (risk) vectors. Once vectors have been chosen for both a detection method and a trojan, a user can use the compare button to perform a comparison between the two vectors. For example, the results of a comparison are shown in the bottom row of Fig. 4. A 1 is displayed when the detection method has a value greater than or equal to the corresponding trojan value, and a 0 otherwise. The zeros in Fig. 4 indicate that the detection method will fail to detect the trojan in the specification phase of the IC life cycle (I_R) and based on the logic type (I_F).

While the detection tool can be employed for individual comparisons, its greatest potential is with the centralized database. Currently the tool only provides comparisons of trojans and detection methods entered by the user. Universal adoption of the HTS will provide a centralized database of all known methods and trojans. This database will allow the detec-

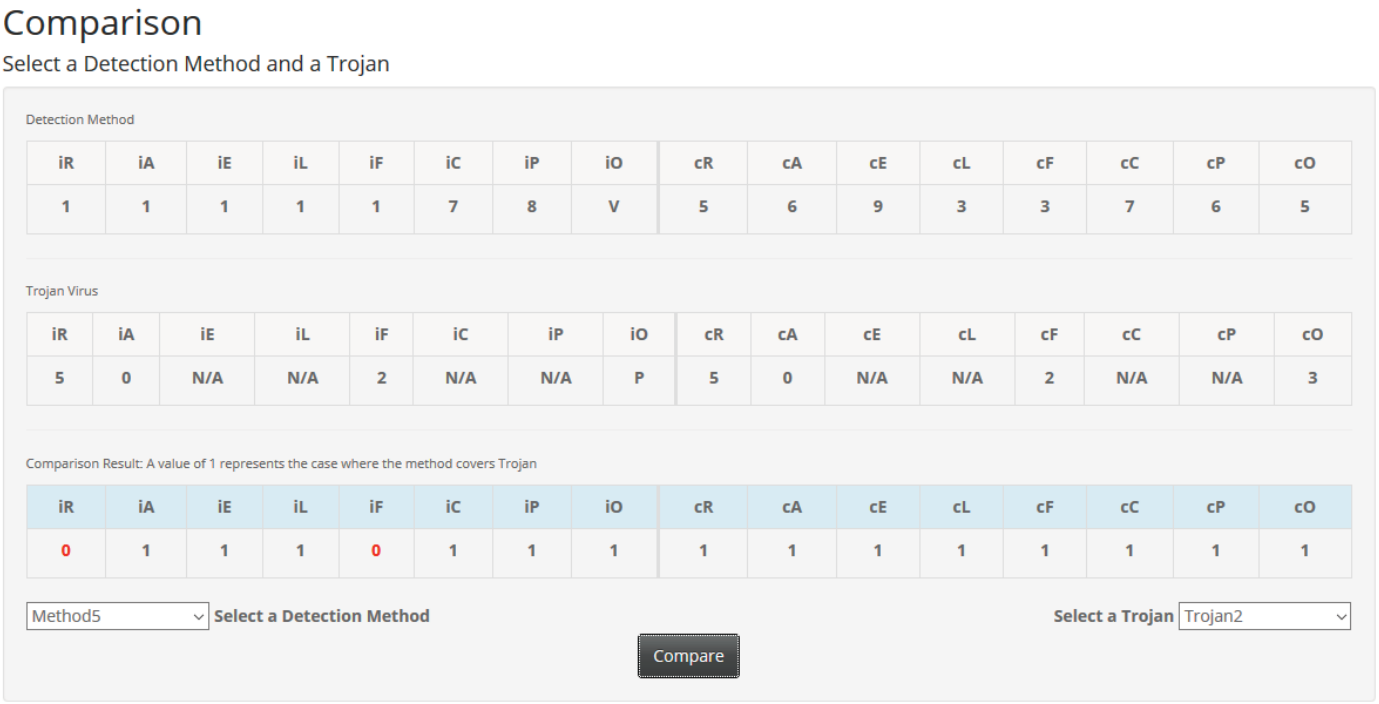


Fig. 4: Detection, coverage and severity vectors.

tion tool to perform comprehensive searches for comparison results. Attackers can use this information to design trojans and defenders can use it to develop solutions for security weaknesses in their designs.

IV. The Web Environment

The hardware trojan detection tool was designed as a web utility for portability and easy distribution. The system is comprised of an application server which performs the computations and generates page markup to minimize the burden on client-side browsers. The server communicates directly with a remote database used to store user login and account information, application data (attributes, categories and hardware trojan matrices). Both the application server and the database are hosted on the *Microsoft Azure Cloud* platform. This platform provides reliability, portability and flexibility with on-demand resources that are automatically managed for scalability and the ability for maintenance to take place anywhere. Fig. 5 shows a block diagram of the automated tool.

The technologies employed are as follows.

- **ASP.NET Web Form:** A user interface focused, event-driven model of the .NET framework. It allows powerful data-binding, separation of server-client side activities, a native security structure, and increased client performance [22].
- **Entity Framework:** An object-relational database mapper designed for the .NET framework. It provides a library of high speed SQL statements wrapped in C# commands to simplify development and ensure performance [23].
- **D3.js:** A Java-script library for visualizing data with HTML, SVG and CSS [24].
- **Azure:** The Microsoft cloud system [25].

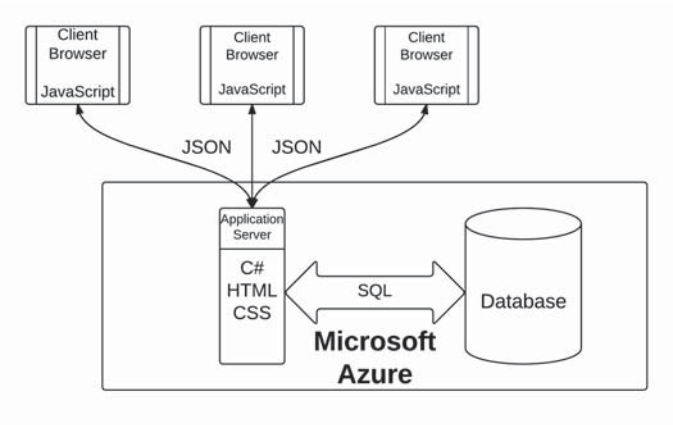


Fig. 5: Block diagram of the hardware trojan automated tool.

Fig. 6 provides an overview of the structure of the trojan system website. The *home, contact*, *about*, and *application information* pages are accessible to all traffic. The application information page contains three sub-pages providing information on each of the primary applications. Users are required to create an account and be logged in to access the remainder of the site. Email confirmation is used to verify user accounts.

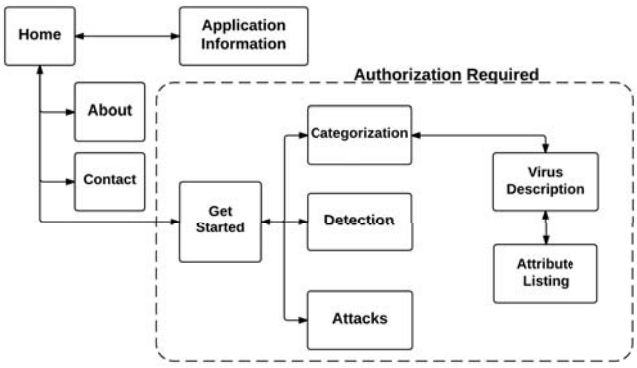


Fig. 6: An overview of the website architecture.

V. Case Study

Consider a circuit designed to compute a function $F(x)$ for a system to authenticate user-password pairs x and $F(x)$. The system performs the arithmetic operation $F(x) = x^2$ to validate users. The use case requires ten users I_0 to I_9 which requires four input bits. The largest function output is 81 meaning seven bits are required for output, Z_1 to Z_7, as illustrated in Fig. 7. A trojan can be inserted into this circuit as shown in Fig. 8 (called a backdoor trojan). The outputs of the original and infected circuits are compared in Table III.

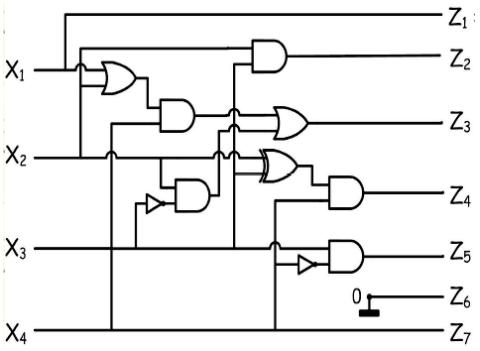


Fig. 7: The trojan free circuit (Circuit A).

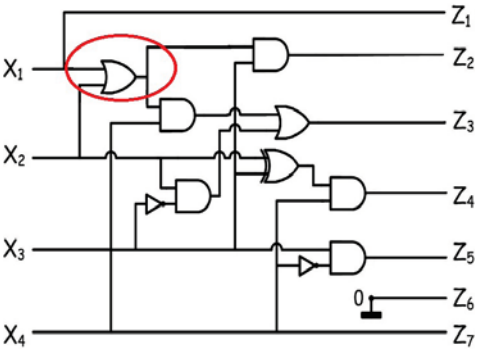


Fig. 8: The backdoor trojan circuit (Circuit B).

TABLE III: Outputs of the Circuits in Figs. 7 and 8 [14]

		Inputs					Circuit A								Circuit B							
		X_1	X_2	X_3	X_4	X	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	$F(x)$	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	$F(x)$
Inputs	I_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	I_1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	I_2	0	0	1	0	2	0	0	0	0	1	0	0	4	0	0	0	0	1	0	0	4
	I_3	0	0	1	1	3	0	0	0	1	0	0	1	9	0	0	0	1	0	0	1	9
	I_4	0	1	0	0	4	0	0	1	0	0	0	0	16	0	0	1	0	0	0	0	16
	I_5	0	1	0	1	5	0	0	1	1	0	0	1	25	0	0	1	1	0	0	1	25
	I_6	0	1	1	0	6	0	1	0	0	1	0	0	36	0	1	0	0	1	0	0	36
	I_7	0	1	1	1	7	0	1	1	0	0	0	1	49	0	1	1	0	0	0	1	49
	I_8	1	0	0	0	8	1	0	0	0	0	0	0	64	1	0	0	0	0	0	0	64
	I_9	1	0	0	1	9	1	0	1	0	0	0	1	81	1	0	1	0	0	0	1	81
Undefined	I_{10}	1	0	1	0	10	1	0	0	0	1	0	0	68	1	1	0	0	1	0	0	100
	I_{11}	1	0	1	1	11	1	0	1	1	0	0	1	89	1	1	1	1	0	0	1	121
	I_{12}	1	1	0	0	12	1	1	1	0	0	0	0	112	1	0	1	0	0	0	0	80
	I_{13}	1	1	0	1	13	1	1	1	1	0	0	1	121	1	0	1	1	0	0	1	89
	I_{14}	1	1	1	0	14	1	1	0	0	1	0	0	100	1	1	0	0	1	0	0	100
	I_{15}	1	1	1	1	15	1	1	1	0	0	0	1	113	1	1	1	0	0	0	1	113

The four input bits of the circuit x_1 to x_4 are used to identify the users I_0 to I_9. A simple test will show that the circuit outputs the desired $F(x) = x^2$ for each of the users. However, upon closer inspection it is noted that the inputs corresponding to $x = 10$ to $x = 15$ are not used. This is a typical vulnerability which can be exploited by an attacker. Inputs $x = 10$ and $x = 11$ to Circuit A produce results which are not correct according to $F(x) = x^2$. This is an intended result used as a security feature for these don't care conditions. If an attacker is able to make the modification in Fig. 8, inputs 10 and 11 will provide correct results according to $F(x) = x^2$, which can be used to allow access to the system by a malicious user. From the previous description of the trojan and referring to Fig. 1, this trojan is comprised of combinational logic (17), its functionality is functional (18), and it is externally triggered (22). Assuming that this trojan was used to create a denial of service (DoS) attack (15), the attributes of this trojan are 15, 17, 18 and 22. The web tool provides an easy to use on-line store style selection when entering input. When the system performs the analysis, a directed graph representation of the

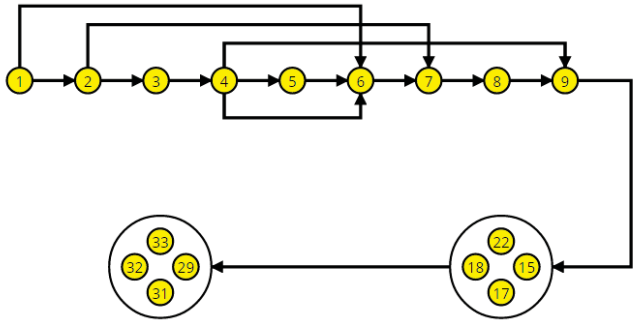


Fig. 9: Classification results based on [21].

result of the classification method in [21] is displayed as shown in Fig. 9. The HTS provides a vector representing the severity of the trojan in which is shown in Fig. 10. There is an option to save the results with a title and description. The user is then

Trojan Virus

iR	iA	iE	iL	iF	iC	iP	iO	cR	cA	cE	cL	cF	cC	cP	cO
5	6	7	1	1	2	3	T	5	6	4	1	1	2	4	4

Fig. 10: Hardware trojan severity results.

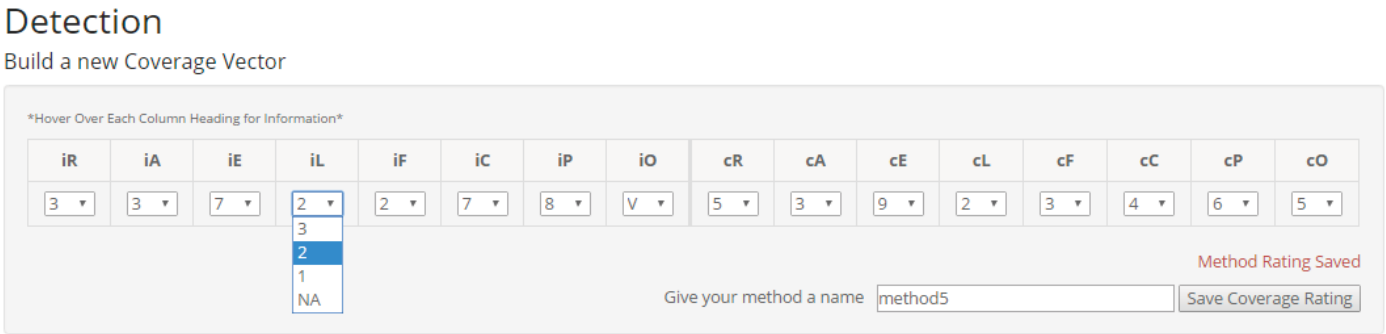


Fig. 11: The coverage vector for a detection method.

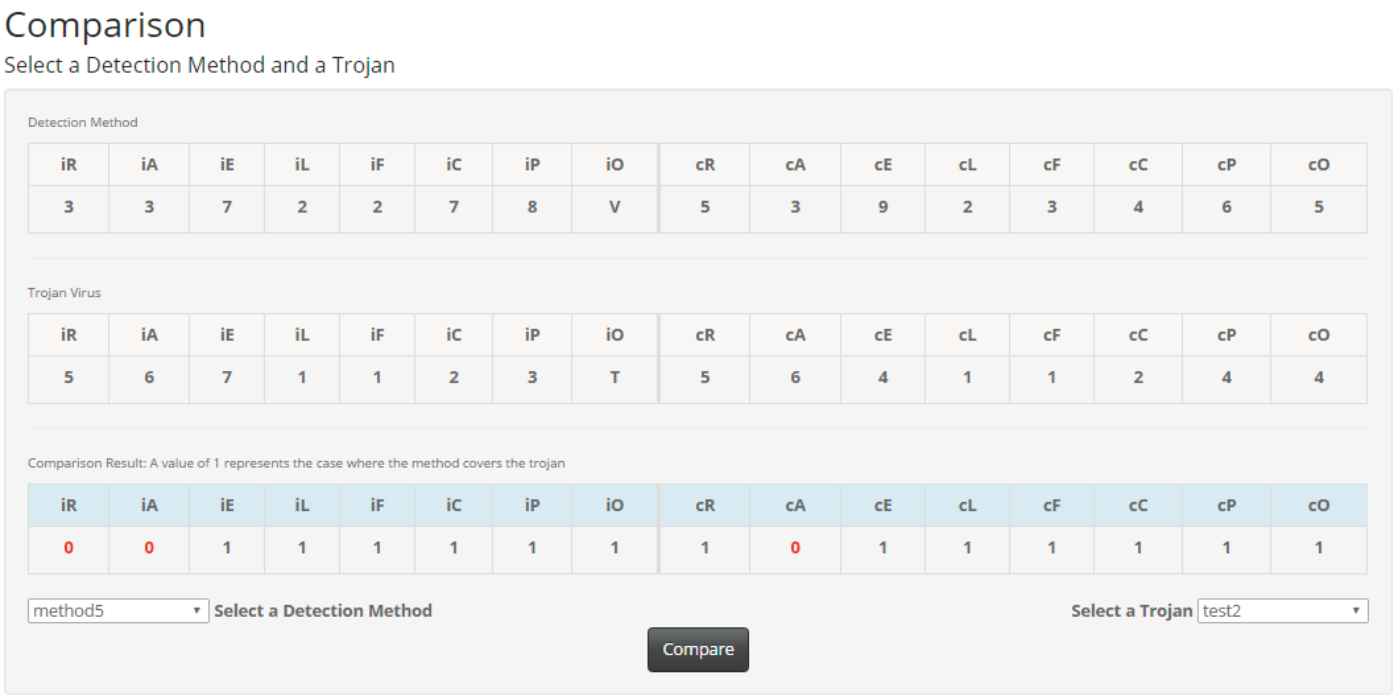


Fig. 12: The case study comparison.

able to move to the detection comparison page shown in Fig. 4. On this page, the saved severity vector can be selected and loaded into the detection tool along with any saved detection method to perform a comparison.

On the detection tool page, a coverage vector for a new detection method was created, as shown in Fig. 11, named *method5*, and saved in the database. The coverage vector for this method was then selected along with the severity for the trojan. Choosing the comparison button provides the results in the bottom row of Fig. 12. The red zeros show the user in which of the eight categories the detection method fails to detect the trojan.

VI. Conclusion

The relatively new field of hardware security has yet to provide a structure that allows developers and security professionals an efficient means of collaboration. The Hardware Trojan System (HTS) is an automated tool that can be employed to simplify the investigation and development of hardware trojans and detection techniques. The universal adoption of this system will provide cohesion to the field of hardware security, and centralizing the techniques in the HTS will expedite adoption. The generation of vectors corresponding to the category attributes for both trojans and detection methods will aid in trojan generation as well as countermeasures. These vectors allow quick and accurate assessment based on the attributes. Defenders concerned with a particular vulnerability can browse the database for an appropriate detection technique. An attacker who has found a weakness in a particular system can search for existing trojans that can be used. The automated tool allows developers to generate and store values which can be used to compare trojans. The severity and coverage vectors are stored in the database along with descriptive information provided by the user. A case study was provided which demonstrates the ease of use of this tool and its usefulness.

The implementation of data mining techniques and an interface that allow users to browse the archive of saved work is left for future work. The addition of this functionality will provide a centralized database of known hardware trojans and detection techniques.

References

[1] C. Marchand and J. Francq, "Low-level implementation and side-channel detection of stealthy hardware trojans on field programmable gate arrays," *IET Computers Digital Tech.*, vol. 8, no. 6, pp. 246–255, Nov. 2014.

[2] Y. Liu, K. Huang, and Y. Makris, "Hardware trojan detection through golden chip-free statistical side-channel fingerprinting," in *Proc. ACM/EDAC/IEEE Design Automation Conf.*, San Francisco, CA, Jun. 2014, pp. 1–6.

[3] Y. Liu, Y. Jin, and Y. Makris, "Hardware trojans in wireless cryptographic ICs: Silicon demonstration & detection method evaluation," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, San Jose, CA, Nov. 2013, pp. 399–404.

[4] D. K. Karunakaran and N. Mohankumar, "Malicious combinational hardware trojan detection by gate level characterization in 90nm technology," in *Proc. Int. Conf. on Computing, Commun. and Networking Tech.*, Hefei, China, Jul. 2014, pp. 1–7.

[5] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware trojan horse detection using gate-level characterization," in *Proc. IEEE/ACM Design Automation Conf.*, San Francisco, CA, Jul. 2009, pp. 688–693.

[6] Y. Cao, C.-H. Chang, and S. Chen, "A cluster-based distributed active current sensing circuit for hardware trojan detection," *IEEE Trans. Inform. Forensics Security*, vol. 9, no. 12, pp. 2220–2231, Dec. 2014.

[7] X. Mingfu, H. Aiqun, and L. Guyue, "Detecting hardware trojan through heuristic partition and activity driven test pattern generation," in *Proc. Commun. Security Conf.*, Beijing, China, May 2014, pp. 1–6.

[8] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *Proc. IEEE Int. Conf. on Hardware-Oriented Security and Trust*, Anaheim, CA, Jun. 2008, pp. 51–57.

[9] M. Li, A. Davoodi, and M. Tehranipoor, "A sensor-assisted self-authentication framework for hardware trojan detection," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition*, Dresden, Germany, Mar. 2012, pp. 1331–1336.

[10] P. Kumar and R. Srinivasan, "Detection of hardware trojan in SEA using path delay," in *Proc. IEEE Students' Conf. on Elect., Electronics and Computer Sci.*, Bhopal, India, Mar. 2014, pp. 1–6.

[11] O. Soll, T. Korak, M. Muehlberghuber, and M. Hutter, "EM-based detection of hardware trojans on FPGAs," in *Proc. IEEE Int. Symp. on Hardware-Oriented Security and Trust*, Arlington, VA, May 2014, pp. 84–87.

[12] A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps," *IEEE Trans. Comput.-Aided Design Integr. Circuits and Sys.*, vol. 33, no. 12, pp. 1792–1805, Dec. 2014.

[13] S. Moein, J. Subramnian, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi, "Classification of hardware trojan detection techniques," in *Proc. Int. Conf. on Computer Eng. and Sys.*, Cairo, Egypt, Dec. 2015, pp. 357–362.

[14] S. Moein, S. Khan, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi, "An attribute based classification of hardware trojans," in *Proc. Int. Conf. on Computer Eng. and Sys.*, Cairo, Egypt, Dec. 2015, pp. 351–356.

[15] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards trojan-free trusted ICs: Problem analysis and detection scheme," in *Proc. Conf. on Design, Automation and Test in Europe*, Munich, Germany, Mar. 2008, pp. 1362–1365.

[16] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware trojans," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, San Jose, CA, Nov. 2008, pp. 632–639.

[17] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *IEEE Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[18] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust*, Anaheim, CA, Jun. 2008, pp. 15–19.

[19] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware trojan horse detection using gate-level characterization," in *Proc. ACM/IEEE Design Automation Conf.*, San Francisco, CA, Jul. 2009, pp. 688–693.

[20] S. Narasimhan et al., "Hardware trojan detection by multiple-parameter side-channel analysis," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2183–2195, Nov. 2013.

[21] N. Houghton, S. Moein, F. Gebali, and T. A. Gulliver, "A web tool for the automation of hardware trojan classification," in *Proc. Int. Conf. on Embedded Systems, Cyber-physical Systems, and Applic.*, Las Vegas, NV, Jul. 2016.

[22] Microsoft (2002) ASP.NET 4.5 [Computer Software] Available: http://www.asp.net/ Accessed: May 7, 2015.

[23] Microsoft (2008) Entity Framework v6.0 [Computer Software] Available: https://msdn.microsoft.com/en-ca/data/ef.aspx Accessed: May 7, 2015.

[24] Microsoft (2011) D3.js v3.5.6 [Computer Software] Available: https://d3js.org/ Accessed: May 7, 2015.

[25] Microsoft (2010) Azure Cloud System [Computer Software] Available: azure.microsoft.com Accessed: May 7, 2015.

ISBN: 1-60132-445-6, CSREA Press ©

SESSION

BIOMETRICS AND FORENSICS

Chair(s)

Dr. Haydar Teymourlouei
Dr. Khaled Ali Shehata

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

Subject Movement at Different Force Levels in a Fingerprint Recognition System

Kevin Chan, Jeffrey Chudik, Katrina Molina, Alex Hirsch, Brennon Morning, Evan Pulliam, Drew Radcliff, Stephen Elliott, Ph.D.
Department of Technology Leadership and Innovation
Purdue University
West Lafayette, Indiana, USA

***Abstract*—There has been an increase in biometric application and advancement, and researchers continuously move to improve the technology. Fingerprint recognition is one of the biometric modalities that has experienced this growth, with its increasing presence in Homeland Security and law enforcement. This study investigated the subject performance movement within a fingerprint recognition system. The performance of a biometric system can be tied to the population using it. Analysis of the population brings context and granularity to performance results. This study analyzed fingerprint data collected by the International Center for Biometric Research (ICBR) back in 2010. DET curves and Zoo plots were gathered and segregated by finger and force. Performance data and error rates of different force level were compared to find the optimum and most meager conditions for each finger.**

***Index Terms*—biometrics, fingerprint recognition, human-computer interaction, zoo menagerie, biometric performance.**

I. Introduction

Biometrics is the automatic identification of individuals using unique physiological and biological traits. Biometrics has traditionally been used by Law Enforcement and Homeland Security but has found increasing traction in private industry [1]. This study aims to look at population factors for continuous improvement on the fingerprint recognition system utilized by these government agencies. Being biological in nature, population factors can contribute to performance variance. Analyzing the population performance under different force levels can add granularity to traditional performance analysis, such as detection-error tradeoffs (DET) and equal error rates (EER) [2].

II. Literature Review

A. Biometrics

Biometrics refers to technologies that measure and analyze human body characteristics for authentication purposes [3]. Biometrics is a type of technology that is used for auto identification and capturing data from users for identification and verification processes relating to their identity and identity management [4]. When biometrics was first being created, the technology available was very limited, it was not until the 1960s when commercial biometric research began. The technology was further advanced and refined in the 1970s and 1980s and then commercialized during the 1990s. Biometrics is now widely used to manage the risk of security breaches and facilitate transactions.

While there are many types of biometrics, each has their strengths and weaknesses. The "best" choice of biometrics is largely dependent on the requirements of the application. The various types of biometric systems can be contrasted based on factors encompassed by factors like distinctiveness, stability, scalability, usability [4].

B. Fingerprint Recognition

A fingerprint recognition system uses an individual's fingerprint scan to identify the user. Originally called Galton points, minutiae are specific locations on a fingerprint that help uniquely identify a fingerprint image, and verify its associated user. These points are ridges, ridge endings, raised portions on the surfaces of the fingers, and bifurcations (a point at which two ridges meet) [4].

Fingerprints can be divided into three separate classifications based upon the ridge patterns that make up a person's fingerprint. These classifications are loops, whorls, and arches. The percentage of the population within each class is not equal, 60-65% of the population has loops, 30-35% has whorls, and the remaining 5-10% has arches. There are subclasses within each class, such as a plain arch versus a tented arch, and each subclass has unique properties that set it apart within the class [4].

In matching classifications, comparisons of the minutiae of a print are used to find a genuine match score and impostor match score. The genuine match is a statistic that measures how well an individual matches against their previously captured scans. An impostor match is a statistic that measures how well an individual can be distinguished from the captured scans of others [3].

C. Biometric Performance

The biometric performance of a population or system is typically measured using various metrics, such as accuracy, efficiency, scalability, and template size [5]. Two

methodologies for measuring performance are DET curves and the Zoo Menagerie [6, 7].

DET (Detection Error Tradeoff) curves examine the system as a whole, and lower FAR and FRR are the markers of an effective system [3, 7]. Normally DET curves are not overlaid, but for comparison in this study, the curves for specific fingers have been overlaid with different force levels. Performance criteria were measured by False Accept Rate (FAR) and False Reject Rate (FRR) [6]. The performance of each finger, at each force level, are indicated in Figures 5 through 12. On a good curve, as the curve progresses, the number of False Rejects (those genuine matches that are not accepted) should decrease, and the number of False Accepts (those imposter matches that are accepted) should decrease.

Figure 7 is a good example of a DET curve that clearly indicates better performance of the system using a particular force level for the left middle finger. Looking at the graph, 9N performs better than any other force level, while 5N performs the poorest until 7N overtakes it. The EER (Equal Error Rate) is also an indicator of the accuracy of the system in regards to the algorithm used to run the matching program. Figure 6 has a wide range of EER scores, indicating that the system for this finger is not as accurate as it should be for the left little finger.

D. Zoo Menagerie

Users of a biometric system have differing degrees of accuracy within the system [7]. Doddington's Zoo was the traditional way of categorizing samples based on verification performance when users matched against themselves and with others [8, 9]. In Doddington's Zoo, the hardest sample to verify was named the goat that did not match well against itself; while the wolf could match well against others (especially lambs), lambs match against themselves, but they also match well with others, making them vulnerable to impersonation [10] [11].

Instead of sheep/goats/lambs/wolves, an additional way to categorize users is to use doves/worms/chameleons/phantoms. These animals are part of Yager and Dunstone's menagerie, which is defined regarding a relationship between the genuine and imposter match scores [10]. This method is more concerned with the dispersion of samples, and whether there are more or less of a particular animal than expected [7]. Yager and Dunstone's animals are mapped on a grid with four colored corners depicting where the animal areas are; there is a top 25%, a bottom 25% and then a combination of the two. Samples are placed in these areas based on statistical performance scores. Those that perform high in both imposter and genuine are chameleons; low imposter and high genuine are doves; high imposter/low genuine are worms; lastly low imposter/low genuine are phantoms. Each animal type should contain approximately 1/16th of the total user population [7].

The performance of a system is important to take into consideration when using a particular method. The zoo plot is a different type of performance indicator than the traditional DET or ROC curve. For a zoo plot representing a population of samples, researchers would want their plot to show that there is the expected 1/16th user population in each area of the graph [8]. In using the zoo plot as a means of analyzing the performance of the system or the population, the researcher need only to consider the skew of the data. If there are a larger than expected number of results in one corner or another, the raw data and images may need to be evaluated for quality and uniformity to be sure there were no methodic issues with the study. The zoo plot is more concerned with the performance of individuals and uses match scores to show performance, rather than with the performance of the population.

III. Methodology

The fingerprint samples analyzed in this study were taken from a previous collection study done in 2010 for the Department of Homeland Security. The study was done by the International Center for Biometric Research (ICBR). ICBR collected data on 154 subjects, and each subject submitted fingerprints at different force levels. The force levels chosen (5 N, 7 N, 9 N, 11 N, and 13 N) were applied on each user's fingerprints using a 10-print device.

Demographic information such as age, ethnicity, and gender were collected from all subjects. The data was categorized by finger, and each finger further subcategorized by force level.

A. Calculation

The fingers used in this experiment were right index, right middle, right ring, right little, left index, left middle, left the ring, left little. To determine the optimal force level of the device, some data including the False Acceptance Rate (FAR), False Rejection Rate (FRR), genuine scores, and imposter scores were analyzed through commercially available biometric matching algorithms.

The analyzed data was then visualized with zoo plots and DET curves with Oxford Wave Research Bio-Metrics 1.5 visualization rendering software. It is important to pay special attention to this data to ensure an efficient system is in place. After testing, the overall quality and efficiency of the system are evaluated, and changes are often made to optimize further the system.

IV. Results

A. Demographics

The study conducted contained 154 individuals. As you can see from the Figure 1 below, which depicts the range of ethnicities within the study, it is clear that the vast majority of the subjects were Caucasian.

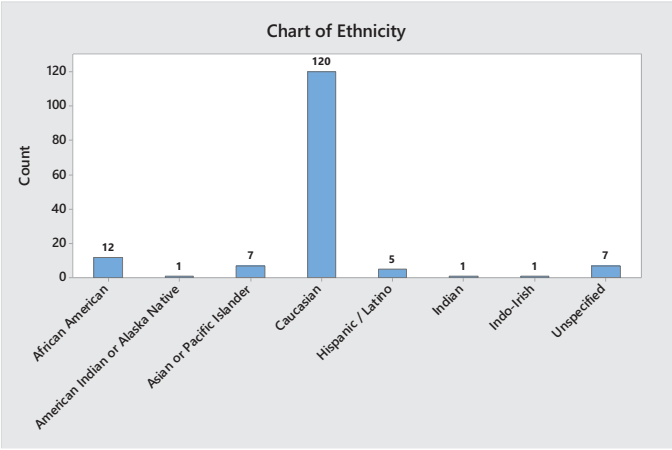


Figure 1. Bar chart of subject ethnicity.

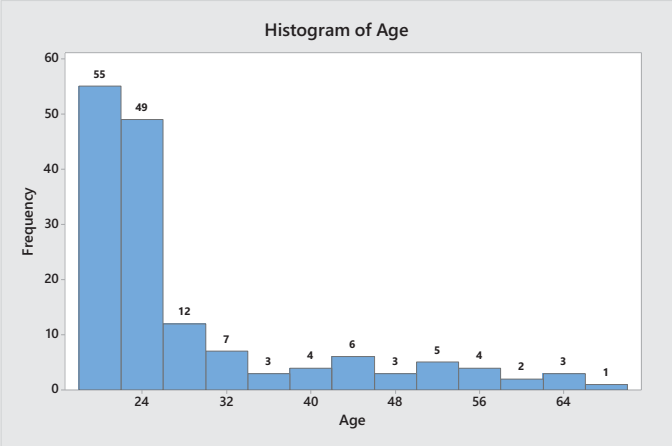


Figure 2. Histogram of subject age.

Figure 2, above, depicts the age range of the subjects in the study. The average of which is a little more than 29 years of age. The majority of subjects, however, were between 20 and 25 years of age. Age data is typically recorded to add context to lower performing fingerprints of older subjects [7].

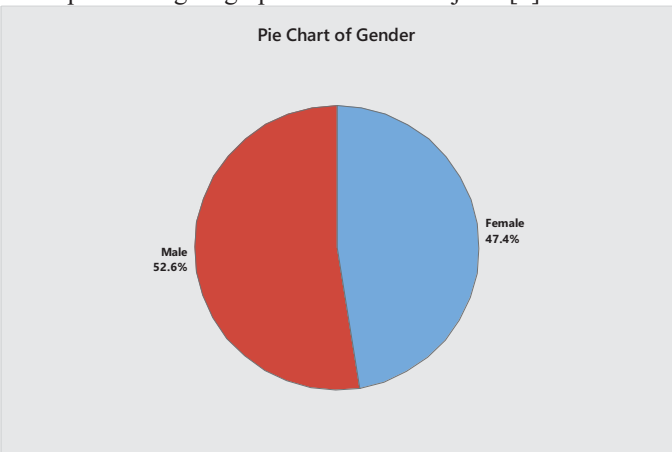


Figure 3. Pie chart of subject gender.

Although the age and ethnicity of all the subjects were heavily skewed towards one direction, Figure 3, above, shows the gender distribution within the study. It was split nearly evenly with a slight majority of subjects being male.

B. *DET Cures and Equal Error Rates*

Table 1 shows the performance results of the fingerprint matching analysis. FRR is shown for logarithmic intervals for of FAR, being 0.01, 0.1, and 1. Figures 4 through 14 and the FRR interval calculations were generated via Oxford Wave.

EER was also recorded for all results. It is important to note that some of the DET curve provided could not display all results due to a logarithmic scale. Figures 6 and 7 only shows four force levels overlaid on a single axis. This was because the error rates for those force levels, 13N on Left Middle and 9N on Left Ring, were too low to appear.

Based on error rates alone, we can see that lower force captures do not perform as well as higher force captures. Index fingers showed lower error rates than the other fingers, with little fingers having much higher error rates.

TABLE 1
PERFORMANCE ANALYSIS

Finger	Force Level	0.01	0.1	1	EER
LI	13	0.87%	0.65%	0.43%	0.5963%
LL	13	4.98%	4.56%	4.11%	3.5820%
LM	13	0.33%	0.00%	0.00%	0.0000%
LR	13	0.00%	0.00%	0.00%	1.0688%
RI	13	1.30%	1.30%	1.26%	1.0162%
RL	13	5.84%	4.75%	3.26%	3.2670%
RM	13	0.22%	0.22%	0.22%	0.2077%
RR	13	0.87%	0.87%	0.80%	0.7312%
LI	11	1.77%	1.73%	1.52%	1.8886%
LL	11	5.19%	3.33%	3.03%	2.8311%
LM	11	1.30%	1.30%	1.30%	1.0436%
LR	11	1.08%	1.08%	1.08%	1.0481%
RI	11	0.22%	0.22%	0.16%	0.1917%
RL	11	2.89%	2.38%	2.16%	1.6495%
RM	11	1.08%	0.87%	0.87%	0.8118%
RR	11	1.95%	1.94%	1.73%	1.7373%
LI	9	0.87%	0.43%	0.43%	0.3103%
LL	9	4.80%	3.68%	2.86%	2.5123%
LM	9	0.22%	0.00%	0.00%	0.0113%
LR	9	0.00%	0.00%	0.00%	0.0400%
RI	9	1.30%	1.30%	1.30%	1.0549%
RL	9	4.55%	3.62%	3.08%	2.6698%
RM	9	1.52%	0.87%	0.65%	0.7099%
RR	9	0.43%	0.22%	0.22%	0.1865%
LI	7	1.08%	0.87%	0.43%	0.5642%
LL	7	11.43%	8.65%	7.25%	6.5041%
LM	7	0.22%	0.22%	0.22%	0.2169%
LR	7	0.22%	0.22%	0.22%	0.2047%
RI	7	1.30%	1.30%	0.79%	0.8630%
RL	7	4.37%	4.11%	3.60%	3.1300%
RM	7	1.73%	1.52%	1.25%	1.0648%
RR	7	1.30%	1.08%	1.08%	1.1466%
LI	5	0.87%	0.65%	0.41%	0.3096%
LL	5	11.26%	8.53%	7.14%	6.5117%
LM	5	2.60%	2.38%	2.16%	2.3496%
LR	5	3.03%	3.03%	3.02%	2.7679%
RI	5	2.81%	2.38%	2.38%	2.1228%
RL	5	6.49%	6.05%	5.42%	4.3585%
RM	5	2.38%	2.38%	1.94%	1.4470%
RR	5	3.03%	3.03%	2.60%	2.3102%

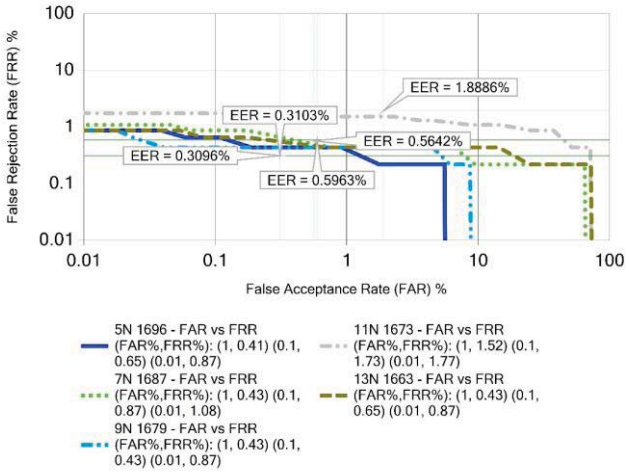


Figure 4. DET Curve of Left Index.

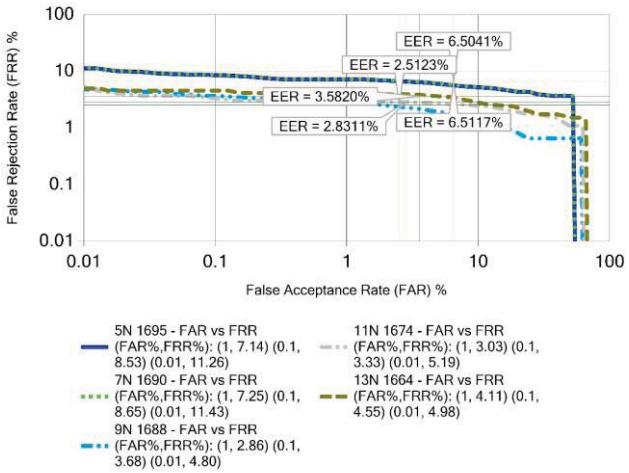


Figure 5. DET Curve of Left Little.

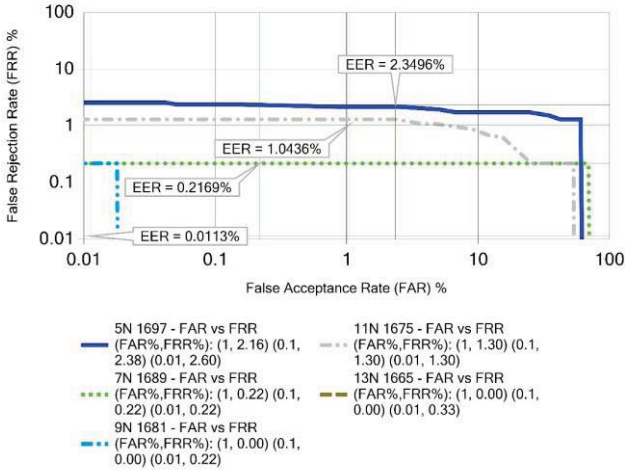


Figure 6. DET Curve of Left Middle.

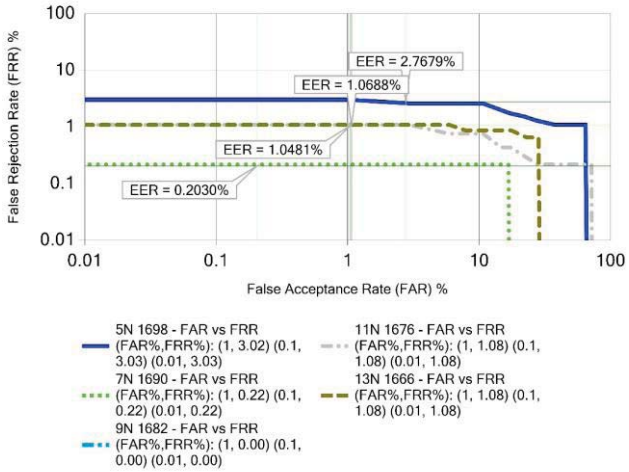


Figure 7. DET Curve of Left Ring.

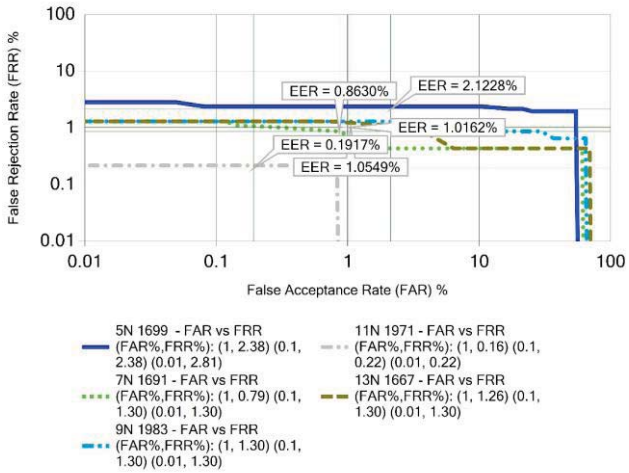


Figure 8. DET Curve of Right Index.

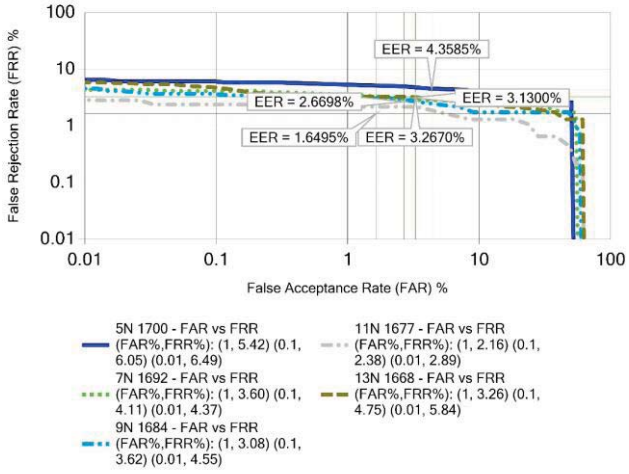


Figure 9. DET Curve of Right Little.

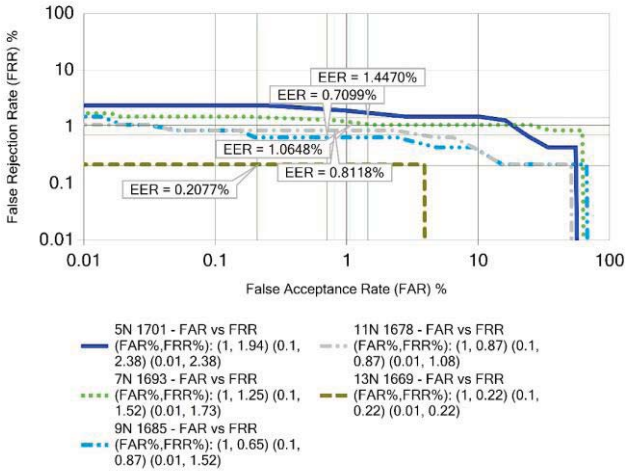


Figure 10. DET Curve of Right Middle.

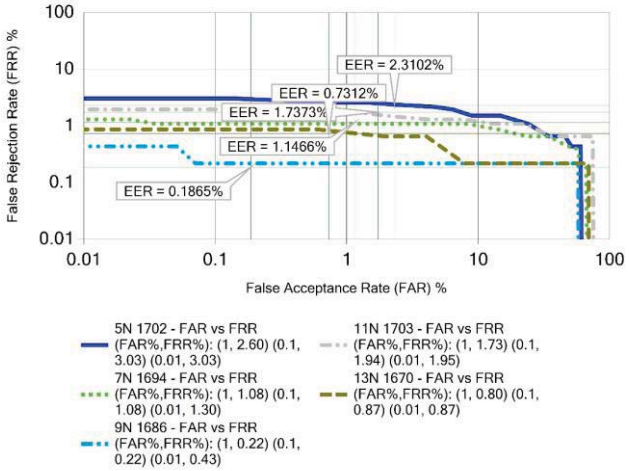


Figure 11. DET Curve of Right Ring.

C. Zoo Plots

TABLE 2
ZOO ANALYSIS RESULTS

Finger	Force Level	Doves	Worms	Chameleons	Phantoms	Normals
LI	13	6	5	12	16	115
LL	13	6	3	15	12	118
LM	13	8	2	13	13	118
LR	13	6	5	18	12	113
RI	13	8	11	12	10	112
RL	13	4	3	18	9	120
RM	13	6	5	16	11	116
RR	13	6	9	13	10	116
LI	11	7	2	10	15	120
LL	11	6	2	16	14	116
LM	11	9	4	14	16	111
LR	11	5	1	13	18	117
RI	11	8	3	12	14	116
RL	11	7	3	19	10	115
RM	11	7	5	14	16	112
RR	11	6	2	14	20	112
LI	9	7	6	11	11	119
LL	9	6	6	12	9	121
LM	9	8	3	15	16	112
LR	9	7	4	14	9	119
RI	9	9	5	18	14	107
RL	9	8	5	12	10	119
RM	9	7	6	15	16	110
RR	9	7	6	15	12	114
LI	7	8	6	12	14	114
LL	7	5	4	14	11	120
LM	7	4	5	13	13	119
LR	7	7	4	14	9	119
RI	7	5	5	15	15	113
RL	7	4	5	18	12	115
RM	7	7	7	14	13	113
RR	7	9	4	17	14	110
LI	5	4	4	19	16	111
LL	5	4	4	19	16	111
LM	5	6	7	13	14	114
LR	5	5	8	16	13	112
RI	5	7	11	10	10	115
RL	5	5	5	14	14	116
RM	5	8	9	12	5	120
RR	5	5	9	11	13	117

V. CONCLUSIONS

The increasing use of biometric technologies in all industries, it is important to understand how different modalities can perform. Knowing how populations perform under different conditions is key to selecting proper testing parameters for maximum success with any population. Our study of fingerprints across multiple force levels allowed us to see clearly the movement of identical samples across different performance indicators and metrics.

A. Zoo Analysis and Animal Movements

Examining the zoo plot results for the left little finger, there is a movement of 10 samples from the "normal" range into the various animal areas due to the decrease in pressure from 9 N (at best performance) to 5 N (worst performance). Not only was there noticeable movement from normal to the chameleon and phantom areas, but even two of the doves move to a less favorable category due to this decrease in pressure. Increasing the pressure from 9 N to 13 N does not impact the doves'

category, but there is a small movement in all the other categories, most notably the worms, moving three samples to other categories through the changes in pressure.

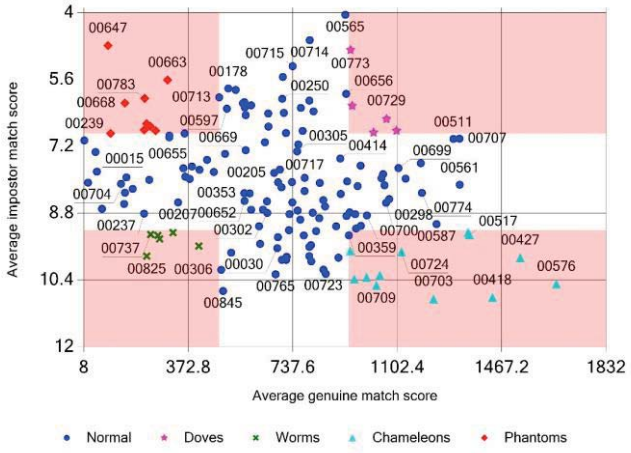


Figure 12. Zoo Plot of Left Little 9N, highest performance.

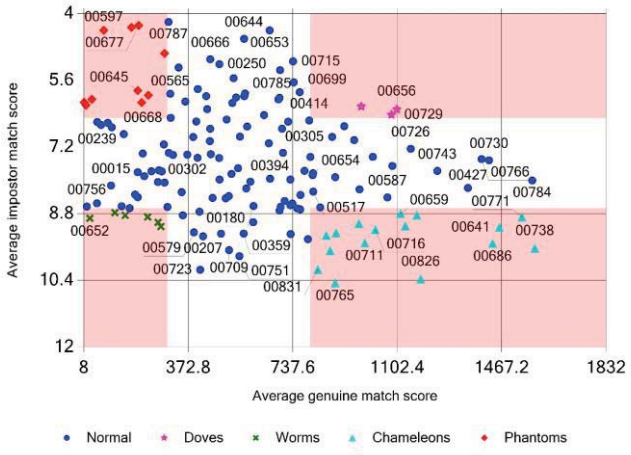


Figure 13. Zoo Plot of Left Little 5N, lowest performance.

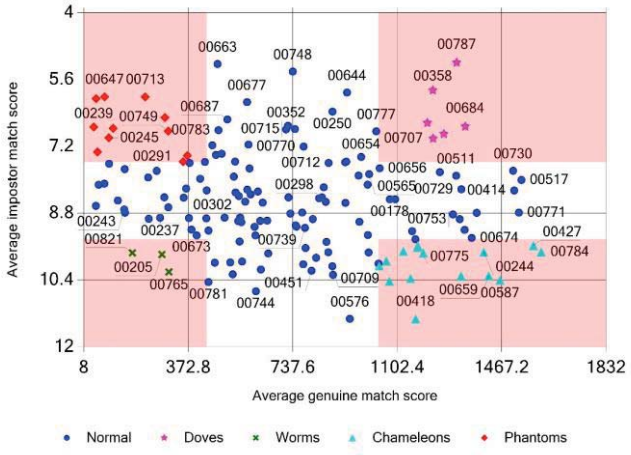


Figure 14. Zoo Plot of Left Little 13N, with little change in animal distribution since 9N.

B. *Analysis of DET Curves*

When considering the DET curve for the left little finger against the other fingers sampled, the performance of this system for the left little finger performs poorly across all force levels, indicating that this system is a poor choice for identification of this population using this finger. In an application of this system, it is important to control for important variables in regards to what the administrator needs to screen their users, and to consider even the system itself as a possible source of variability.

In the analysis of data, the zoo plot allows for quick visual inspection of results, and if more than one study is conducted, the variables and their effects on the samples can be more easily compared to each other if all other variables are controlled. Likewise, the DET curve allows for quick review of system performance in the case of a skewed set.

In real-world implementation and application of this and other systems, it is important to remember that biometric systems are not perfect. As seen from the movement of samples throughout the different force levels, there are consistency issues that have not yet been addressed.

Overall all fingers increase in genuine and impostor scores when the force level is increased. Little fingers perform poorly overall, with the lowest EER of 1.65% for Right Little at 11N. The best performance recorded was from the Left Middle, with EERs of 0.01% at 9N and 0.00% at 13N. The most consistent performance recorded was from the Right Middle, with no EERs above 1.5%. In respects to all fingers, there was no practical improvement to EERs at 13N. After 9N, there were no practical benefit to increasing the force level. Little and Ring fingers actually had a resurgence of EERs when force was increased from 9N to 11N and 13N.

C. *Ongoing Research*

Moving forward it will be important to look into why samples change and how this can be controlled or taken into account to better construct studies for known populations.

Our study was based on a narrow range of ages and ethnicities, to improve the systems used and gain more information, it will be necessary to widen the sample population with follow-on studies to include more diversity.

Additionally, habituation to the collection device and process, as well as the quality of the images collected should also be analyzed. Over time performance of the sample population may increase because placement of the finger and pressure level achievement is easier, and the distortion of the three-dimensional finger in the two-dimensional image will become more uniform. This may not be true of an unhabituated population.

Further inspection of the zoo menagerie would be the next stage in performance analysis for this research. Now that the approximation of genuine and impostor distribution is recorded, genuine and impostor stability scores can be index. This would further define which subjects have instances of performance instability, and allow researchers to analyze original fingerprints to connect physiological and biological characteristics to the reported performance.

VI. References

[1] A. K. Jain, A. Ross and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 14, no. 1, pp. 4-20, 30 January 2004.

[2] K. O'Connor, S. Elliott, M. Sutton and M. Dyrenfurth, "Stability of Individuals in a Fingerprint System across Force Levels – An Introduction to the Stability Score Index," in *The 10th International Conference on Information Technology and Applications*, 2015.

[3] A. K. Jain, A. A. Ross and K. Nandakumar, Introduction to Biometrics, New York, New York: Springer Science & Business Media, 2014.

[4] L. Hong and A. Jain, "Classification of fingerprint images," in *Proceedings of the Scandinavian Conference on Image Analysis*, 1999.

[5] D. Maltoni, D. Maio, A. K. Jain and S. Prabhakar, Hanbook of Fingerprint Recognition, Springer Science & Busniess Media, 2009.

[6] T. Dunstone and N. Yager, Biometric system and data analysis: Design, evaluation, and data mining, Eveleigh, New South Wales: Springer Science & Business Media, 2008.

[7] N. Yager and T. Dunstone, "The Biometric Menagerie," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 32, no. 2, pp. 220 - 230, 2010.

[8] M. E. Schuckers, Computational Methods in Biometric Authentication, M. Jordan, R. Nowak and B. Schölkopf, Eds., New York: Springer Science & Business Media, 2012.

[9] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," National Institute of Standards and Technology, Gaithersburg, 1997.

[10] T. Dunstone and N. Yager, "Worms, Chameleons, Phantoms and Doves: New Additions to the Biometric Menagerie," in *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, Alghero, 2007.

[11] G. Doddington, W. Liggett, A. Martin, M. Przybocki and D. Reynolds, "Sheep, Goats, Lambs and Wolves: A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation," National Institute of Standards and Technology (NIST), Gaithersburg, MD, 1998.

A Multimodal Biometric System with Several Degrees of Feature Fusion for Target Identities Recognition

Sorin Soviany [1], **Cristina Soviany**[2], **Sorin Puşcoci**[1]
[1]T.C.T. Department , I.N.S.C.C., Bucharest, Romania
[2]Feature Analytics, Bruxelles, Belgium

Abstract -*This paper presents an innovative design for multimodal biometric systems with several degrees of feature-level fusion. It is a hierarchical feature-level fusion allowing to efficiently exploit the informative properties of the features that are extracted from several independent sources. A focused application is the accurate recognition of the target identities for mobile individuals that use their smartphones and other devices with resources-related constraints, for example to login to their accounts and do online banking on their mobile devices. The proposed feature-level biometric fusion method requires homogeneous feature vectors in order to perform their functional combination and to avoid the concatenation-based fusion*.

Keywords: multimodal, feature-level fusion, target identity

1 Introduction

In the multimodal biometrics data from several human traits are combined in order to improve the individual recognition accuracy. Several independent traits of the same person provide a higher confidence level in the accuracy of recognition process, giving the intrinsic performances of the integrated biometrics. The higher security is ensured because it is more difficult to simultaneously fake several biometric samples of the enrolled users.

The multimodal biometric systems are typically based on fusion rules that are applied on the input data within various processing stages. There are 2 main categories of biometric fusion rules, depending on the processing stage: pre-classification fusion rules and post-classification fusion rules.

The pre-classification fusion includes sensor-level and feature-level fusion. The post-classification fusion includes matching score-level, rank-level and decision-level fusion [1]. This is the most implemented fusion because of its low complexity; the fusion process is applied within an advanced stage of data processing and therefore it does not very efficiently exploit the most informative properties of the extracted features. The feature-level fusion is assumed to be more promising for further improvements in recognition accuracy. However its implementation is more challenging comparing with the post-classification fusion; this is due to the complexity of various feature extraction algorithms, the incompatibilities among the various features sets and the curse of dimensionality problem.

We propose a multi-level feature fusion method able to more efficiently exploit the informative properties of the features. The method is applied on a bimodal (fingerprint and iris) biometric system and includes 2 levels of feature fusion: a local fusion for each biometric and a global fusion for both biometrics. The generated feature vectors are homogeneous because for both biometrics the feature extraction algorithms are quite similar; the only differences are given by the parameterization of the feature extractors. The classification module is designed to perform the target identities recognition, according to the real applications requirements. The focus is on mobile applications with several security degrees in which not all the authorized users have the same permissions to access the protected resources.

The remainder of this paper is structured as follows. Section II presents some related works in the area of feature-level biometric fusion. Section III presents the proposed feature-level method. Section IV presents the experimental results. Section V concludes our research.

2 Related works

The feature-level fusion was approached for several biometrics and feature extraction methods. The typical feature fusion rules are the concatenation and the weighted averaging of the features, respectively, with their own assumptions for application [1].

Basically the feature-level fusion could be performed in 2 ways [2]. One way is to separately apply a feature extraction algorithm for each biometric; the resulted features are directly fused using concatenation or a functional combination, depending on the feature vectors homogeneity degree with care about the incompatibility among the features. In the 2nd way the original images are fused and a single feature extraction algorithm is applied to obtain the overall feature vector containing the information from the original fused biometric traits.

In [3], the feature-level fusion of hand and face was applied in 3 scenarios: fusion of PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) coefficients of face; fusion of LDA coefficients from the R, G, B components of a face image; fusion of face and hand. The

fusion rule in all these cases was the concatenation of the feature vectors.

Another multimodal system with feature-level fusion (face and palmprint) was presented in [4]. The pre-processed data were sequentially transformed for dimensionality reduction and uncorrelating (with PCA) and for a better class discrimination (with LDA). The resulted feature vectors were concatenated. The performance improvement was ensured by the sequence of feature space transformations PCA and LDA.

A cryptographic application of multimodal biometrics with feature-level fusion of fingerprint and iris was proposed in [5]. The goal was to generate secure cryptographic keys based on multimodal biometrics. The fusion procedure of fingerprint and iris data involved the following operations: shuffling of individual feature vectors; concatenation of shuffled feature vectors; merging of the concatenated feature vectors. A cryptographic key was generated from the fused features.

Another application of feature-level fusion was described in [6]. The feature-level fusion was designed in order to re-arrange and combine 2 binary biometric templates aiming to efficiently exploit the error correction capacities. This procedure was applied for the iris biometric.

Despite of the previously mentioned challenges, the feature-level biometric fusion has a lot of promising applications in cryptosystems design and development. Another example was given in [7], in which the fused feature vector resulted from concatenation of the individual binary strings; the fused vector was applied to generate a secure key for the whole authentication process.

The feature-level fusion was also applied for other biometrics, for example palm veins and signature [8]. Here the fusion was realized by feature concatenation, which is the most applied fusion scheme, especially for heterogeneous feature vectors.

The feature fusion of palmprint and iris was approached in [9] with a wavelet-based fusion scheme. The required assumptions were the same-sized images and their association with the same colour.

A PCA based image features fusion method for iris and face was proposed in [10]. The PCA transform was used to provide the best representation of the input data, enabling to apply a functional combination of the 2 biometrics, finally resulting in a fused image from which the individual recognition had to be performed.

Another multi-biometric fusion for identity authentication used a strategy based on dual iris, visible and thermal face traits [11]. Initially a feature-level fusion was applied on the features that were previously extracted from each biometric. Then a matching score fusion was also applied for the matching scores of iris and face. The feature-level fusion was based on a functional combination of the extracted and selected features.

The different fusion rules were applied also for unimodal biometrics, in order to increase the recognition accuracy. An example was given in [12] for iris recognition with feature and score fusion based multiple classifier selection. The input data were provided from left and right iris. The feature-level fusion was performed through concatenation of the features extracted from the left and right iris.

Other biometrics used for multimodal feature-level fusion design were ear and retina [13]. In this model, the feature-level fusion was done using an additive functional combination of the features extracted from both ear and retinal images.

These are only a few examples for biometric feature-level fusion and its applications. This fusion is still a challenge, despite of its promising advantages in recognition accuracy improvement. The main challenges are [1]: to find out some relationships among the feature spaces of various biometrics; the curse of dimensionality associated with the feature vectors concatenation; the unavailability of the feature vectors for the commercial biometric systems.

Here we propose a feature-level fusion method in which for several biometrics we apply a local and a global feature fusion. The feature vectors for each biometric are homogeneous and it is not mandatory to increase the dimensionality using the concatenation-based feature-level fusion.

3 The feature-level fusion method

3.1 The bimodal biometric system architecture with feature-level fusion

The bimodal (fingerprint+iris) biometric system architecture is depicted in figure 1. The basic operations are

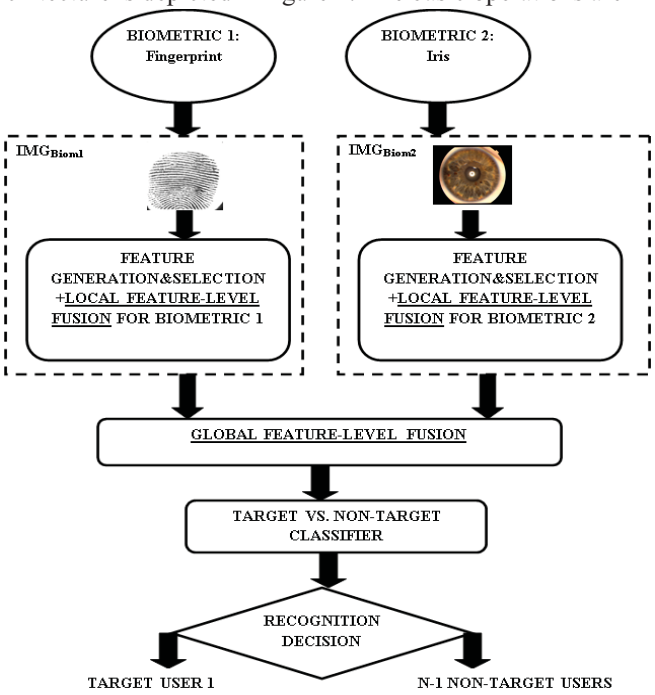


Figure 1 : The bimodal system architecture

the following:

- biometric samples acquisition for both traits ;
- feature generation/selection with the *local feature-level fusion* processes for each biometric ;
- *global feature-level fusion* of both biometrics ;
- target-vs.-non-target classification applied on the overall fused biometric feature vectors ;
- the recognition decision.

The biometric samples are provided from N=20 mobile individuals.

Next we describe the feature generation, fusion operations and the classification process.

3.2 The feature generation and local feature-level fusion for each biometric

For the feature extraction/generation we use a regional approach with textural features exploiting the 2nd order statistical properties of the images pixels, given from co-occurrence matrices. The overall procedure for feature extraction and *local feature-level fusion* is depicted in figure 2. The *local feature-level fusion* is the combination process of several feature sub-vectors that are generated for each biometric; the *local fusion* is separately applied for fingerprint and iris extracted features, respectively.

First we perform the *manual selection of the regions of interest* (ROIs) from each of the input images. The ROIs

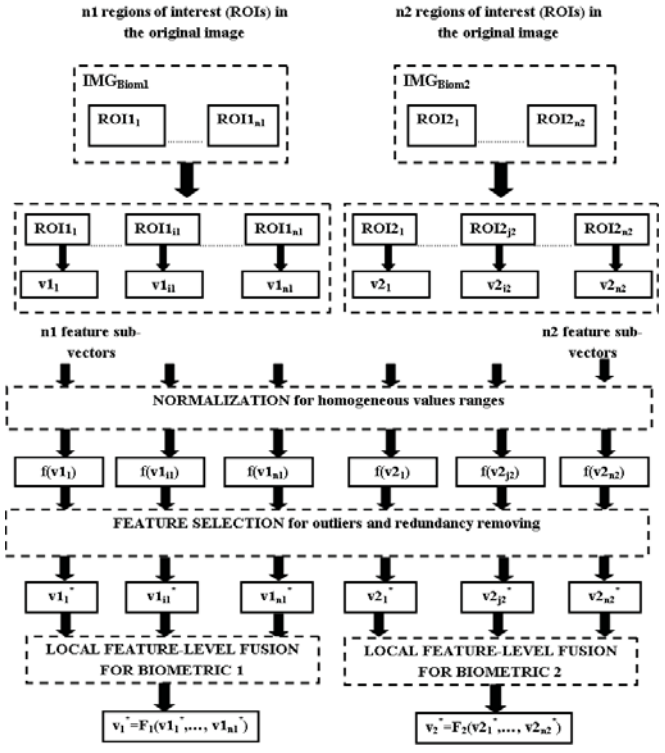


Figure 2 : The feature generation and local feature-level fusion

number is n_k, where k is the index of the corresponding biometric trait (1 for fingerprint and 2 for iris). Here: $n_1 = n_2 = 4$. This operation is performed applying mask matrices on the original input images representation. We extract features from only 4 ROIs per image. The goal is to work with small-dimensional feature spaces ensuring a reduced computational complexity; this is because the application is the recognition of mobile individuals using devices with resources-related constraints. Before this we convert the original color images to gray-scale images with the method given in [14], reasoned by its simplicity.

The feature computing is based on 2nd order statistics in each of the selected ROIs, working with co-occurrence matrices. These features show the gray levels distribution within the input image [15], [16]. The feature sub-vectors are directly derived from the co-occurrence matrices computed for each ROI. The advantage of the *co-occurrence matrices-based feature extraction* is that it efficiently captures the underlying texture of the original image [16], [17]. The co-occurrence matrices computation for each ROI uses the following amounts [16], [18]:

- *GLB*, *the number of gray-level bins*. $GLB_1=6$ (for fingerprint image ROIs) and $GLB_2=8$ (for iris image ROIs). This provides many significant values and less 0 values in the co-occurrence matrices, to get informative textural features for each ROI;
- *OFFSET, the displacement distance*, the number of pixels between the pixels pairs used to compute the co-occurrence matrix. This should not exceed a certain threshold, otherwise the overall number of pixel pairs will be small and therefore the resulted useful information will be poor. In our case the optimal value is *OFFSET=2* for both biometrics

The *GLB* vs. *OFFSET* ratio allows to easily adjust the resulted feature space dimensionality, meeting the low-complexity mobile application requirements.

On the other hand, the same feature extraction algorithm used for both biometrics provides homogenous feature vectors and this is why the feature-level fusion could be performed with a functional combination of the vectors, avoiding their concatenation and the curse of dimensionality problem. The concatenation feature fusion avoidance is also supported by the same sized feature sub-vectors that are derived from this ROI-based extraction feature parameterization.

For each biometric we get $n_k = 4$ feature sub-vectors ($k = 1$ for fingerprint and $k = 2$ for iris): $vk_{i_k}, i_k = \overline{1, n_k}$ is the feature sub-vector derived from the ROI i_k selected within the original biometric image k.

The co-occurrence matrices feature extractor parameterization provides the following sizes of the resulted feature sub-vectors:

$$l(v1_{i_1}) = \left(GLB_{i_1}\right)^2 = l_{1,0}, i_1 = \overline{1, n_1} \quad (1)$$

$$l(v2_{j_2}) = \left(GLB_{j_2}\right)^2 = l_{2,0}, j_2 = \overline{1, n_2} \quad (2)$$

The resulting feature sub-vectors sizes, before the normalization and feature selection, are: $l_{1,0}=36$ (for fingerprint) and $l_{2,0}=64$ (for iris).

Next we apply a *normalization* process to all these feature sub-vectors in order to have common ranges of the feature values. The normalization operation ensures a certain homogeneity degree of the resulted features. The process is realized with the sigmoid function computing values within the range [0,1], according to:

$$f\left(vk_{i_k}\right)=\frac{1}{1+\exp\left(-\left(A_{k,i_k}\cdot vk_{i_k}+B_{k,i_k}\right)\right)}, k=\overline{1,2}, i_k=\overline{1,n_k} \quad (3)$$

where: vk_{i_k} is the feature sub-vector generated for the biometric k using the ROI i_k; $n_k = 4$ is the number of ROIs selected within each of the input original images; A_{k,i_k} and B_{k,i_k} are the scaling and offset parameters of the sigmoid function. The values ranges are given for the available experimental data, all of them being within 1 and 2.5. The slightly higher limits are achieved for iris features.

The next step is *feature selection* for outliers and redundancy removal. It provides a further dimensionality reduction but keeping the most informative features for the individual recognition. We evaluate several feature selection procedures using as criterion the 1-NN (nearest neighbor) rule, because of its asymptotic property of classification error rate limitation [19]:

$$\varepsilon^* \le \varepsilon_{1-NN} \le 2\cdot\varepsilon^*\cdot\left(1-\varepsilon^*\right)\le 2\cdot\varepsilon^* \quad (4)$$

in which: ε^* is the optimal Bayes classifier error rate; ε_{1-NN} is the 1-NN classifier error rate. This property shows that the 1-NN rule is asymptotically at most twice as bad as the Bayes decision rule.

The tested non-optimal and non-exhaustive feature selection algorithms are the following: *forward-searching*, *backward searching*, *individual ranking*, *random feature selection* and *floating-search*. The best execution time is provided by individual ranking feature selection for all the cases.

The resulting normalized feature sub-vectors for fingerprint and iris, together with their sizes, are the following: $vk^*_{i_k}, k=\overline{1,2}, i_k=\overline{1,n_k}$ and

$$l\left(v1^*_{i_1}\right)=l^*_1, i_1=\overline{1,n_1} \quad (5)$$

$$l\left(v2^*_{j_2}\right)=l^*_2, j_2=\overline{1,n_2} \quad (6)$$

At this stage we have the following sizes of the generated feature sub-vectors for the 2 biometrics: $l^*_1=18$ for each of the 4 fingerprint feature sub-vectors; $l^*_2=30$ for each of the 4 iris feature sub-vectors.

The next operation is *the local feature-level fusion* separately for each biometric [20],[21], actually an *intra-modal fusion*. The fusion is based on a functional combination of the generated feature sub-vectors, and not on their concatenation. This is supported by the same size and the homogeneity of the feature sub-vectors. We have 2 functions generating the final feature vectors for fingerprint and iris:

$$v1^*=F_1(v1^*_1,...,v1^*_{n_1}) \quad (7)$$

$$v2^*=F_2(v2^*_1,...,v2^*_{n_2}) \quad (8)$$

The suitable models for local feature fusion are based on the weighted average of the feature sub-vectors. The computational simplicity is the main reason. The final feature vectors for fingerprint and iris are given by

$$vk^*(m_k)=\frac{\sum_{i_1=1}^{n_k} w_{i_k}(m_k)\cdot vk^*_{i_k}(m_k)}{\sum_{i_k=1}^{n_k} w_{i_k}(m_k)}, k=\overline{1,2}, m_k=\overline{0,l^*_k-1} \quad (9)$$

This is the *local (intra-modal) feature-level fusion rule* in which: w_{i_k} is the weights set associated to the feature sub-vector $vk^*_{i_k}$; k is the biometric index (1 for fingerprint, 2 for iris); m_k is the index for the individual feature; l^*_k is the size of each feature sub-vector.

The *local (intra-modal) feature-level fusion* combines several homogeneous feature sub-vectors derived from the same biometric with a certain parametrized function, while avoiding the concatenation-based feature-level fusion and the curse of dimensionality problem for the further classification.

3.3 The global feature-level fusion for both integrated biometrics

The *global (inter-modal) feature-level fusion* is the process in which 2 feature vectors derived from different biometrics are combined into a single feature vector. This fusion is typically done by the feature vectors concatenation. This approach increases the dimensionality leading to the curse of dimensionality [15]. In our method *the global (inter-modal) feature-level fusion* is performed with a functional combination of the 2 different feature vectors, but with some particular transformations (figure 3).

At the beginning of the *global feature-level fusion* process we have the 2 separate feature vectors resulting from the local *feature-level fusion stage*, $v1^*$ and $v2^*$ with their sizes l^*_1 and l^*_2: $l(v1^*)=l^*_1=18$ and $l(v2^*)=l^*_2=30$. The goal is to achieve the same feature space size supporting a functional combination-based fusion.

Firstly we apply a *feature space transform for dimensionality reduction* (*PCA - Principal Component Analysis*, if only the simple feature decorrelation is required, or *ICA – Independent Component Analysis*, for the features statistical independence). We retain the same number of best uncorrelated (independent) features for both biometrics: $n_{P/I,1}=n_{P/I,2}=n_{P/I}=12$. This is the common size for the both feature vectors, providing the dimensionality reduction. PCA is an unsupervised transform and we apply it on the following weighted covariance matrix S computed for C classes:

$$S=\sum_{i=1}^{C} f_i\cdot S_i \quad (10)$$

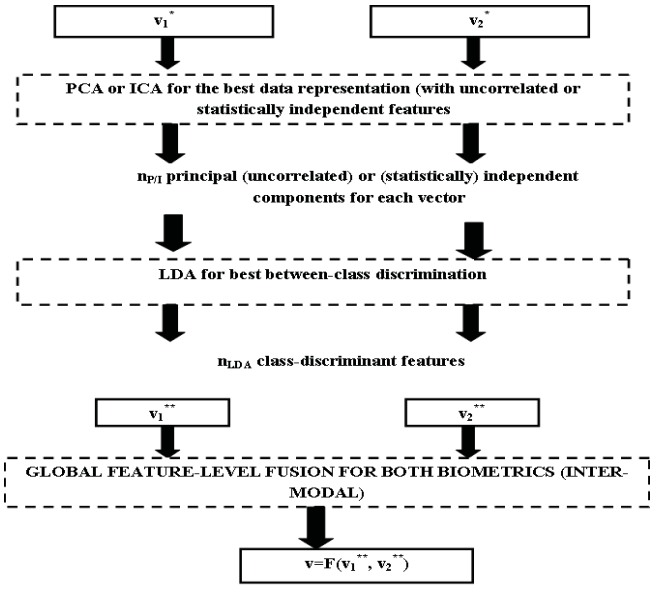


Figure 3 : The global feature-level fusion

in which: S_i is the class covariance matrix of the biometric samples belonging to the class i; f_i is the representation degree of class i within the training set (relative frequency). In a typical identification process there is an one-to-one mapping class: enrolled identity, sometimes with an additional class for an unknown (non-registered) individual. However, the application concerns end-users with several authorization degrees; this is why we consider the following $C = 2$ biometric classes: class 1 with the biometric samples from the most important end-user and class 2 including data from the other $N – 1$ users.

The next step is the *LDA (Linear Discriminant Analysis) transform* to select the best class-discriminant features. This leads to the optimal linear transformation w that maximizes the Fisher Discriminant Ratio (FDR) [15]:

$$FDR(w) = \frac{w^T \cdot S_B \cdot w}{w^T \cdot S_W \cdot w} \qquad (11)$$

S_B and S_W are the scatter matrices between-class and within-class, respectively. The resulting optimal features number for the both biometrics is:

$$n_{LDA,1} = l(v_1^{**}) = n_{LDA,2} = l(v_2^{**}) = n_{LDA} = 8 \qquad (12)$$

v_1^{**} and v_2^{**} are the resulting fingerprint and iris feature vectors, respectively.

Finally, having the same reduced size of the feature vectors, we can do *the global (inter-modal) feature-level fusion*. The fusion is based on a functional combination of the 2 feature vectors, the weighted average, like for the *local feature-level fusion*, due to its low computational complexity:

$$v = F(v_1^{**}, v_2^{**}) = \frac{w_1 \cdot v_1^{**} + w_2 \cdot v_2^{**}}{w_1 + w_2} \qquad (13)$$

The weights $w_k, k = \overline{1,2}$ are properly selected according to the performance of each biometric.

` We do not use the concatenation-based fusion, so the final fused feature space dimensionality is reduced allowing to avoid the curse of dimensionality within the stage of classification for people recognition.

3.4 The classification

The data classification for identification requires grouping the biometric samples from one person into one class. Our focus is a multi-level security application, so we group the biometric samples from the most important user into one class (class 1) and the data from the other $N – 1$ users into the 2nd class (class 2), reducing the multi-class problem to a binary classification one.

The small number of fused features (8) allows to apply a low-complexity classification model, LDC (Linear Discriminant Classifier). This is suitable for a small-sized feature space and for a training dataset size significantly exceeding the feature space size, to avoid the peaking and curse of dimensionality issues.

The underlying discriminant function $g(v)$ is given by:

$$g(v) = (\mu_1 - \mu_2)^T \cdot S_0^{-1} \cdot v + const \qquad (13)$$

where v is the fused feature vector (fingerprint+iris);
μ_1 and μ_2 are means vectors of the 2 classes;

$$S_0 = P_1 \cdot S_1 + P_2 \cdot S_2 \qquad (14)$$

S_0 is the weighted covariance matrix that is required because LDC is suitable for Gaussian distributions with equal class covariance matrices. The weights are the class prior probabilities P_1 and P_2. S_1 and S_2 are the class covariance matrices;

$$const = -\frac{1}{2} \cdot \mu_1^T \cdot S_0^{-1} \cdot \mu_1 + \frac{1}{2} \cdot \mu_2^T \cdot S_0^{-1} \cdot \mu_2 + \log\left(\frac{P_1}{P_2}\right) \qquad (15)$$

The training set has 20 to 50 biometric samples per class. This range allows the resulting ratio *training set size (Ts)/feature space size (Fs)* to prevent the peaking and curse of dimensionality. This ratio should be within [2, 10] range [15]. The reliable generalization capacity of the classifier is provided with a leave-one out cross-validation procedure.

4 Experimental results

The performance of the bimodal biometric system with 2 degrees of feature-level fusion (*intra-modal* and *inter-modal*) is evaluated for several training set sizes and a fixed fused feature space size ($Fs = 8$), to find out the optimal *Ts/Fs* ratio. This value should ensure the curse of dimensionality and peaking avoidance for a design with low-complexity classifier and a resources-constrained application. The performance measure is True Positive Rate (TPR) for class 1 (the most important user) for which the recognition accuracy should be maximized. The performance is evaluated for various classification rejection thresholds. The rejection threshold allows to adjust the system performance according to the

input data quality. Figure 4 depicts the TPR vs. Rejection Thresholds for the bimodal system.

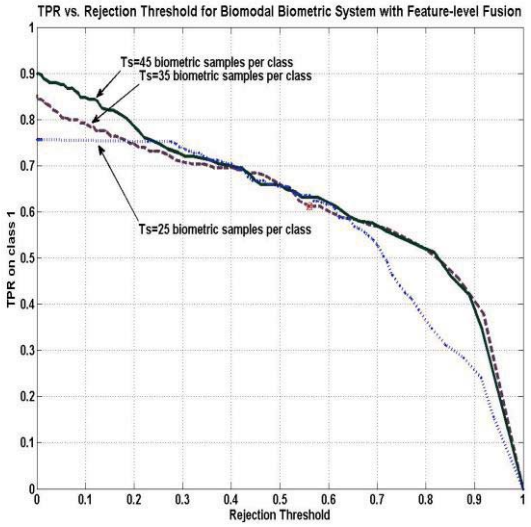


Figure 4: TPR vs. Rejection Threshold for the Bimodal System

Actually the *Ts/Fs* ratio could be optimized with the following varying conditions: varying *Ts* and fixing *Fs*; varying *Fs* and fixing *Ts*; varying *Ts* and varying *Fs*. In this ongoing research we only applied the 1st condition for the moment.

One can see the improvement in class 1 recognition accuracy for several training samples per class. This is the recognition for the target individual with the highest authorization degree. The performance improvement decreases with the rejection threshold increasing. In biometric applications the classification rejection threshold should be fixed between 1 and 10%, typically around 5%; this is the typical fraction of individuals that are not able to provide suitable biometric samples. The rejection threshold prevents to classify low-quality biometric data and it improves the recognition accuracy to more than 97% which is a target for high-level security applications such as online banking on mobile devices. However this performance target strongly depends on the biometric traits intrinsic accuracy ; not all the human traits provide the same inherent performance for individuals recognition. The input data quality is adjusted with a suitable parameterization of the co-occurrence matrices-based feature extractor. The lower quality of input data could be compensated with a suitable setting of this threshold but also with an optimal *Ts/Fs* ratio. The rejection threshold should not be increased too much otherwise the system performance will quickly fall down. Therefore a fixed threshold of 5% is reliable for a suitable performance improvement.

The experimental results concerning the recognition accuracy are achieved for the following sizes of the training sets (*Ts*): 25, 35 and 45 biometric samples per class, respectively. The reasons are related to the effects on the *Ts/Fs* ratio on the overall system behavior. This ratio drives the complexity issues in classifier design, taking in account the peaking and curse of dimensionality. The fused feature space size is reduced to 8 dimensions (*Fs = 8*) and the training set size is suitable for a low-complexity classifier (LDC in our design), such as not exceeding 50 samples per biometric class. The class unbalancing problem remains a subject for further research. The achieved ratios *Ts/Fs* are within the required ranges for preventing peaking. However, one can see that the optimal *Ts* values should be 35 or 45 samples per class in order to provide the best recognition accuracy (*TPR > 0.8*) for a rejection threshold of 5%.

These results are achieved for a very small fused feature space dimensionality comparing with other related developments. Most of them are based on the feature vectors concatenation; the applied feature extraction algorithms in those cases generate large feature spaces despite of some feature selection. In our approach the feature extraction is done with the same algorithm for both biometrics, resulting in a certain homogeneity degree of the feature vectors. This allows a feature-level fusion based on the functional combination of the same-sized feature vectors.

5 Conclusions

We defined a feature-level biometric fusion with 2 layers: a local (intra-modal) fusion and a global (inter-modal) fusion. In both cases the fusion was done with a functional combination of the feature vectors, provided by their homogeneity and common size. The homogeneity was given by the same co-occurrence matrices-based feature extraction algorithms that are applied for both biometrics (fingerprint and iris).

We reduced the resulted overall feature space size and the required training samples number, in order to minimize or at least to reduce the chances of peaking and curse of dimensionality for a low-complexity classifier (LDC). The experimental achievements concerned the optimal training set size vs. feature space size optimal ratio, by varying the first parameter for a fixed overall dimensionality. The overall dimensionality was reduced to only 8 fused features using PCA and LDA transforms on the fused feature vector. The multi-degree feature-level fusion provided the performance improvement for this lower feature space, also avoiding the concatenation fusion and thereby the curse of dimensionality issues.

Further researches concerning the feature-level fusion as a reliable way to increase the recognition accuracy should consider various degrees of features homogeneity. Also the training set size remains a challenge for multimodal biometric systems design.

6 References

[1] Jain A.,Nandakumar K.,Ross A.: "Score Normalization in multimodal biometric systems, Pattern Recognition, The Journal of the Pattern Recognition Society, 38 (2005).

[2] Zhang D.,Song F.,Xu Y.,Liang Z.: "Advanced Pattern Recognition Technologies with Applications to Biometrics", Medical Information Science Reference, IGI Global, 2009.

[3] Ross A.,Govindarajan R. : "Feature Level Fusion Using Hand and Face Biometrics", Proc. of SPIE Conference on Biometric Technology for Human Identification II., Vol. 5779, Orlando, USA, March 2005.

[4] Ahmad M.I.,Woo W.L.,Dlay S.S.: "Multimodal Biometric Fusion at Feature Level: Face and Palmprint", 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP), Newcastle, U.K., July 21-23, 2010.

[5] Jagadeesan A.,Duraiswamy K.:"Secured Cryptographic Key Generation From Multimodal Biometrics: Feature Level Fusion of Fingerprint and Iris", International Journal of Computer Science and Information Security (IJCSIS), Vol. 7, No. 2, February 2010.

[6] Rathgeb C.,Uhl A.,Wild P.:"Reliability-balanced Feature Level Fusion for Fuzzy Commitment Scheme", Proceedings of the 1st International Joint Conference on Biometrics (IJCB'11), Washington, USA, October 10-13, 2011.

[7] Nagar A.,Nandakumar K.,Jain A.:"Multibiometric Cryptosystems Based on Feature-Level Fusion", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 1, February 2012.

[8] Soliman H.,Mohamed A.S.,Atwan A.:"Feature Level Fusion of Palm Veins and Signature Biometrics", International Journal of Video & Image Processing and Network Security IJVIPNS-IJENS Vol. 12, No. 01 28, 2012.

[9] Gayathri R., Ramamoorthy P.: "Feature Level Fusion of Palmprint and Iris", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 1, July 2012.

[10] Nair S.A.H.,Aruna P.,Vadivukarassi M.:"PCA based Image Fusion of Face and Iris Biometric Features", International Journal on Advanced Computer Theory and Engineering (IJACTE), Vol. 1, Issue 2, 2013.

[11] Wang N., Li Q., El-Latif A.A.A., Peng J., Niu X.: "Multibiometrics Fusion for Identity Authentication: Dual Iris, Visible and Thermal Face Imagery", International Journal of Security and Its Applications, Vol. 7, No. 3, May 2013.

[12] Islam R.: "Feature and Score Fusion Based Multiple Classifier Selection for Iris Recognition", Hindawi Publishing Corporation, Computational Intelligence and Neuroscience, 2014.

[13] Tyagi S.K, Kumar J., Singhal N.:"New improved Multimodal Biometric Recognition System: Taking Ear and Retina as Biometric Traits", International Journal of Technical Research and Applications", Vol. 3, Issue 3, May-June 2015.

[14] Bhattacharyya D.,Das P.,Bandyopadhyay S.K., Kim T.: "IRIS Texture Analysis and Feature Extraction for Biometric Pattern Recognition", International Journal of Database Theory and Application, vol. 1, nr. 1, pp. 53-60, December 2008.

[15] Theodoridis S., Koutroumbas K.: "Pattern Recognition"4th edition, Academic Press Elsevier, 2009.

[16] Soviany S., Soviany C., Puşcoci S.:"An Optimized Iris Recognition System for Multi-level Security Applications", The 2014 International Conference on Security and Management (SAM'14), World Academy of Science, Las Vegas, SUA, July 21-24, 2014.

[17] Zucker S.W., Terzopoulos D.: "Finding Structure in Co-Occurence Matrices for Texture Analysis", Computer Graphics and Image Processing nr. 12, 1980.

[18] Bino S. V, A. Unnikrishnan and Kannan B.: "Gray level Co-Occurrence Matrices: Generalization and some new features", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.2, No.2, April 2012.

[19] Devroye L., Gyorfy L., Lugosi G.: "A Probabilistic Theory of Pattern Recognition", Springer, 1997.

[20] Soviany S., Soviany C.:"A Biometric Security Model with Identities Detection and Local Feature-level Fusion", The 2013 International Conference on Security and Management (SAM'13), World Academy of Science, Las Vegas, SUA, July 22-25, 2013.

[21] Soviany S., Puşcoci S.:"A Feature Correlation-based Fusion Method for Fingerprint and Palmprint Identification Systems", The 4th IEEE International Conference on E-Health and Bioengineering – EHB 2013 Grigore T. Popa University of Medicine and Pharmacy, Iaşi, Romania, November 21-23, 2013

Digital Forensic Analysis of Web-Browser Based Attacks

Sally M. Mohamed[1], **Nashwa Abdelbaki**[1], **and Ahmed F. Shosha**[1]
[1]School of Communication and Information Technology, Cairo, Egypt

Abstract— *In recent years, attacks that target browsers' vulnerabilities have increased significantly. An innocent user may lure to access untrusted website and malicious content passively downloaded and executed by his/her web browser; this attack vector known as, Drive-by-Download attack. Systems and security researchers addressed this attack from different perspectives. Several techniques and tools were introduced to detect and prevent Drive-by-Download attack; however, few research addresses the browser forensics perspectives to (1) identify traces (2) reconstruct the executed events of a downloaded malicious content, to assist the digital forensic investigation process. In this paper, digital forensic method is introduced to investigate a web browser subject to Drive-by-Download attack. A Proof-of-Concept implementation based on Firefox browser-extension was developed to inspect and analyze malicious URLs that host malicious executable. The developed system was tested using 55 malicious web pages and successfully identified the digital evidence of the attack. 77% of the identified evidence were artifacts that we believe it could assist forensic investigator to determine if web-browser or a system subject to examination is compromised or not, and the indications of compromises.*

Keywords: Web Browser forensics, JavaScript based attack, Drive-By-Download, Malware, Postmortem Analysis

1. Introduction

Nowadays, users and corporates are more and more connected to the web. A user can access his sensitive business/non-business applications using a web-browser. The web browsers, however, are complex software that developed using various technologies and have to process different file formats and contents that may be vulnerable or contain malicious code. On the other hand, cybercriminals understand that the user is the weakest link in the security chain, and a higher possibility to a successful attack. That's why attackers are trying to exploit vulnerabilities in web-browsers or luring users to visit malicious websites. In a typical Drive-by-Download attack, an innocent user is redirected to malicious web page, commonly denoted as (landing site[1]) [1]. This page contains code (often written in JavaScript[2]), that exploits a browser's vulnerability; browser's installed plug-ins or insecurely designed APIs. If succeeded, the exploits will download a malware from a malicious site[3] into the victim's machine. Usually a Drive-by-Download attack is developed for a specific vulnerability in a specific browser's version, so a common initial activity in this attack vector is reconnaissance and fingerprinting the web-browser metadata. An embedded script will attempt to collect information about the browser type, version, language, installed plug-ins and the installed operating system. Based on the collected information a malicious shell code will download the appropriate exploit or it may behave in a completely benign manner if, i.e. an analysis environment detected [2].

In order to understand the anatomy of the attack, the infected machine's web browser has to be forensically examined. *The Browser Forensics* is an emerging topic of the digital forensic science that refers to the process of extracting and analyzing the web-browsers artifacts and the user's browsing activities for forensic investigation purposes [3]. It is a technology-dependent domain that focuses on the most popular and commonly used web-browsers, i.e. Chrome browser developed by Google, Mozilla's Firefox browser and Internet Explorer by Microsoft, Safari browser developed by Apple Inc; some less commonly used browsers may also be considered, such as Text-Based web-browser i.e. Lynx Viewer. These browsers store a significant amount of data about the user's activities over the Internet if it's used in its normal mode and less data may also be collected if user opt to browse in the private browsing mode [6][7]. The data provided by the web-browsers are not exactly the same [8]. In order to collect these data, one has to consider the following:

- How a web-browser stores data and in which format?
- What are the minimum basic information that can be found in each browser?
- What are the additional and relative information provided by each browser?

All the major browsers would contain information about the browsing history, web applications' cache, web cookies,

[1]**Landing site:** page contains the shellcode (small binary payload) that will exploit vulnerability in the user's browser/plug-in.

[2]**JavaScript:** is a portable language(once written, it can be executed on any browser with JavaScript support). JavaScript attracts both developers and attackers by its dynamic and flexible features, that's why it's so popular and most Drive-by-Download attacks are implemented using it.

[3]**Malicious site:** page contains the malware downloaded by the shellcode. Often the browser will be targeted by a chain of redirection operations before getting to the malicious site.

user bookmarks, form completion data, stored passwords and many more. For example, Mozilla's Firefox and Google's chrome use a SQLite database to store these data, while Microsoft's IE uses files (like index.dat, cache and cookies files) to store it. Additionally, a considerable amount of data can also be found in the installed browser's extensions[4].

Although these browsers store a lot of data about the user's activities over the Internet, still a digital forensic investigation process is required to reconstruct the browser activities (i.e. executed code from a URL, downloaded resources, etc) resulted after accessing a malicious URL. To reconstruct the attack executed events and analyze its actions, we developed a system using Mozilla Debugger API to monitor, log and debug the details of an executed malicious JavaScript code subject to investigation. Output data after analyzing a list of digital forensics evidence are produced based on the following categorization:

- **Volatile Evidences:** non long lasting digital evidences that are residual in a system CPU, processor caches, and/or system memory.
- **Non-volatile Evidences:** digital evidences that are residual on the file system.

In summary, the contributions of this paper are defined as follows:

1) Digital forensic analysis methodology is proposed to allow forensic examination of Web-browsers artifacts, mostly enabled by executing malicious JavaScript code embedded in a malicious Web page.
2) A Proof-of-Concept Implementation of a system based on Firefox browser extension that outputs a digital forensic trace file for the executed JavaScript code. That includes, a detailed information about the volatile and non-volatile forensic evidences resulted from the malicious JavaScript code execution.

The remainder of this paper is structured as follows: Section Two presents the related work. Section Three introduces the proposed forensic analysis methodology. Section Four presents the preliminary analysis and results. Finally, Section Five concludes the paper and defines the possible future work.

2. Related Work

A Drive-by-Download attack could be defined as malicious content downloaded to a user's system without his/her consent using the web-browser. This content may be in different file format, i.e. it can be a malicious Flash file or embedded action script code [9], malicious pdf with embedded JavaScript code [10] or obfuscated JavaScript code in a web page [11] that exploits a vulnerability in the user's system. These downloads can be triggered by different actions, i.e. opening, scrolling or hovering a mouse cursor over a malicious web page or iframe. Academic and professional researches are commonly focusing on the detection and prevention techniques of this attack vector. The currently proposed techniques are mainly based on either analyzing the properties of a malicious web page URL [1] or analyzing the JavaScript code (or any other present code) contained in the web page using a (1) static, (2) dynamic, or (3) a combination of the both, aka, hybrid analysis, as follows:

- **Static analysis:** it uses a set of predefined features to determine that a malicious pattern or code exists in particular web page without code execution; several machine learning techniques and approaches may also be integrated in the static analysis to (1) define the set of features required for the analysis, (2) cluster, classify, and/or determine malicious web pages out of benign web pages [4] [12]. In this analysis approach, a low processing overhead may be required; however, the static analysis generally can be impeded if an obfuscation and/or encryption methods are employed [13].
- **Dynamic/semi dynamic analysis:** It uses a controlled environment, commonly called Sandboxing, to execute a subset or all of the possible execution paths for the embedded code to detect the presence of a malicious behavior. In this approach, a malicious code can be monitored accurately through the execution process. However, additional processing resources may be required. Attackers are also developing their malicious code to detect the analysis environment and to execute a legitimate code to mislead a human analyst or suppresses the execution and attempts to self-delete the code [4] [14].
- **Hybrid analysis:** It uses a combination of both static and dynamic analysis for the embedded JavaScript code to detect Drive-by-Download attacks, and to avoid the drawbacks associated with each approach. Typically, a static analysis technique is used as an initial filter to define the web pages that require a dynamic analysis, i.e. to determine the code that may defeat the dynamic analysis. Applying a hybrid analysis may guarantee accurate detection with minimum resources [4][15].

Other researches focus on analyzing exploit kits that are used to launch a Drive-by-Download attack [5]. *Exploit kit* is a malicious toolkit that exploits security flaws found in software applications. Using an exploit kit requires no proficiency or software development background, and it is equipped with different detection-avoidance methods. This justifies the notable wide-usage of exploit-kits not only by skilled cybercriminal, but also, by a non-skilled malicious users to achieve their purposes. In [20], the authors focuses on the server side part of a Drive-by-Download attack. They analyzed the source code for multiple exploit kits using Pexy, which is a system for bypassing the fingerprinting of an

[4] **A browser extension** is a program that extends the browser functionality and adds extra features to it.

exploit kit and get all of its possible exploits by extracting a list of possible URL parameters and user agents that can be used.

In a recent study presented in [16], the authors propose a system using Chrome JavaScript Debugger to detect browser's extensions that inject malicious ads into a web page. The study revealed that 24% of ad networks domain bring malicious ads. These ads will redirect the user to a landing page, which will finally download a malware/malicious executable into the user's machine. On the forensics side, researches alike [6][7][17] focus on private/portable browsing and how to collect/find the remaining evidences from the memory and the file system. In [18], the authors talked about the importance of making an integrated analysis for different browsers at the same time to understand what happened. They also propose a tool for constructing a timeline for the user's activities. There are a lot of commercial/free browser forensics tools, that give investigators an insight into user's browsing history but none of them deals with the browser memory.

3. The Proposed Digital Forensics Methodology

In this section, we propose our digital forensic methodology to investigate a malicious web page that is suspected to download and further execute malicious code using a web browser in a system subject to investigation. A typical scenario would be: a user notices uncommon activities occur in his system such as suspicious web advertisements appeared while surfing the web. Admins in a corporate network system may notice unusual network traffic inbound or outbound from the compromised system or visiting a web server known to host malicious contents. In these cases, a forensic analyst would perform an examination to the system to determine indication of a compromise, such as searching for a URL to malicious web pages in web-browsing history, a cookie file or a temp file in the Internet storage directory. If identified, it is crucial for the forensic investigation to determine what other resources had been downloaded and executed into the browser from this malicious website.

To approach a forensic analysis for web-browsers subject to any variant of web-based attacks, our proposed methodology consists of the following sequential procedures:

1) **Data Gathering:** it is the process of accessing the malicious URL in a setting similar to a compromised system, to lure the malicious URL to download the set of resources (content, code, and exploit payload) similar to those have been downloaded in the system subject to investigation.
2) **Data Analysis:** it is the process of executing, debugging the code downloaded from the malicious URL and producing a forensic trace file for objects, operations, functions created and/or called from the code. The trace file is further analyzed to extract all of the possible forensic evidences that would assist the investigation.
3) **Data Classification:** itis the process of classifying the forensic traces into subsets that could or could not assist the investigation. For example, traces would be classified into volatile, non-volatile forensic evidence, and traces that would not support the investigation, such as script files embedded by Google-ads that are almost exist in all websites. These script files may be downloaded along the malicious content but it is not relevant to the attack and cannot be considered in the forensic analysis process. The output from the classification process not only can be used to identify evidences in the compromise system, but also, can be used to develop a signature for the attack to locate attack traces in other systems in the network (postmortem analysis).

Figure 1, depicts a visual description for the proposed forensic methodology.

3.1 Data Gathering

The process of data collection requires simulating the settings of a compromised system subject to investigation to avoid downloading and executing code that has never been executed in the original system subject of the incident. In this scenario, an assumption has been made that the user was running a Firefox web-browser. As such, Firefox Browser Extension was developed to monitor, log, and debug the downloaded resources after accessing a malicious web page with a particular attention to the executed embedded JavaScript code. In the Proof-of-Concept implementation, Mozilla Debugger API[5] was used to develop a browser extension that outputs a detailed trace file. This trace file logs and lists the code executed from accessing the page subject to the forensic investigation. The trace file generated in a JSON[6] file format that includes objects created/accessed/-modified on the system with details about the stack frames of the executed code and the execution timestamps. The JSON object contains the following items:

- Type: a string describing the executed frame ('call', 'eval', global', 'debugger')
- Class: a string describing the ECMAScript [7] class of the referent
- Function name: the name of the function whose application created this frame (null if this is not a 'call' frame)

[5]**Mozilla Debugger API:** is a debugging interface provided by Mozilla JavaScript engine "SpiderMonkey", which enables JavaScript code to observe and manipulate the execution of other JavaScript code.

[6]**JSON:** JavaScript Object Notation

[7]**ECMAScript** is a trademarked scripting language specification standardized by ECMA International in ECMA-262 and ISO/IEC 16262.

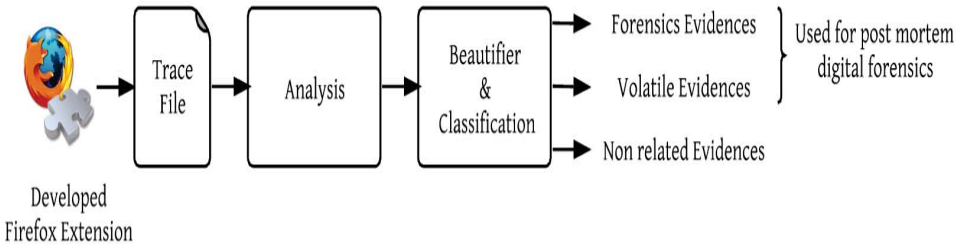


Fig. 1: A visualization for a web-browser forensics system

- Parameters: An object with the name/value pairs of the passed parameters to the called function
- URL: The url of the page in which the function have been called
- Script: The script being executed in this frame

We tested the browser extension using number of real world malicious URLs (55 malicious web site were analyzed) collected from public malware databases[8].

3.2 Data Analysis

The analysis for the generated trace files requires a detailed examination for the extracted JavaScript code. We developed an analyzer using NodeJS[9] to search for specific patterns using regular expressions. We search for patterns of obfuscation events, encoding/decoding events, checking for vulnerability events, URL redirection events, downloading external resources, creating local files on the system, etc. There are different well-known and commonly used techniques for cybercriminals to use a JavaScript code to perform malicious actions. For example, to download an external resource, an attacker may employ one of the following methods:

- Create a script tag and set the source attribute to the required downloadable file:

```
(function() {
var as =
document.createElement('script');
as.type = 'text/javascript';
as.async = true;
as.src =
 "xxxd31qbv1cthcecs.
 cloudfront.net/atrk.js ";
var s =
 document.
  getElementsByTagName('script')[0];
s.parentNode.insertBefore(as, s);
})();
```

- Create an image tag with source to a malicious URL:

```
var tempImage = new Image();
tempImage.src =
smf_prepareScriptUrl(smf_scripturl)
+ 'action=keepalive;time=' + curTime;
```

These different methods were taken into consideration during the analysis process. The output from the analyzer shows the number of occurrence or usage for each event. After this we transformed the extracted code into a human readable format, this is by utilizing a web-based services named JavaScript beautifier[10]. The extracted code was further analyzed to get a closer look into each evidence and manually extract the common patterns between various URLs.

3.3 Data Classification

To avoid providing a forensic analyst with a significant amount of irrelevant information, data classification and analysis procedure is a crucial activity for eliminating data that is not relevant to the case subject to investigation. As stated in [19], a forensic evidence has to characterize the following:

1) Admissible in the court
2) Authentic: the evidence proves a specific action in a specific incident
3) Complete/Exculpatory: can be used to support or refute a user action
4) Reliable: the procedure used for collecting and analyzing the evidence must guarantee the evidence validity and authenticity

The above characteristics and the aforementioned properties of the Drive-by-Download attack were considered in the evidences classification process. If evidence is related to the examined attack type, it will then be classified as relative and is further classified into (1) Volatile or (2) Non-volatile forensic evidence. Other data will be considered as none related evidences. For example:

- Volatile evidence: i.e. in memory shell-code, encoding/encryption code.

[8] www.malwaredomainlist.com/mdl.php, http://www.malwareurl.com/

[9] NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

[10] http://jsbeautifier.org, http://codebeautify.org/jsviewer

Table 1: Properties list example

Evidence Name	Properties
URL redirection	domain name, path
Vulnerability check	branches depth, vulnerabilities name
String manipulation	string operations, string value and length
Downloaded resource	resource name, source url, resource type
Created file	file name, file path, file type

- Non-volatile evidence: Created file on the system, downloaded resource, URL redirection with a trace of the URL in the browser history, etc.

The main reason behind this classification is to ensure that a forensic analyst would know that a volatile data related to the investigation may present but not necessary can be recovered nor reconstructed, and only the identified non-volatile forensics can be further used to develop an attak signature. The forensic analyst can then use the generated attack signature to detect if there is an attack on other systems.

After manually classified identified evidences, a list of properties will be extracted based on the evidence type, to reveal more information about it. Table 1 lists the possible properties for each evidence based on its type.

In addition, the following code sample for example shows a cookie file created from one of the analyzed JavaScript files.

```
f.cookie = e + "=; expires=
Thu, 01 Jan 1970 00:00:01 GMT;
path=/ " + (t ? "; domain= " + (l("msi ") ?
 " " : ". ") + "xxxaddthis.com " : " ")
```

Most of the analyzed URLs were also checking vulnerabilities and were trying to create an ActiveX object or shockwave flash which is an Adobe's Flash Player built directly into the browser as shown below.

```
d = "ShockwaveFlash ";
...d = b[c], -1 < d[r][q]("Shockwave Flash ")
&& n(e = d.description[y]("Shockwave Flash ")
[1]);
....
try {
  c = new ActiveXObject(d + ".7 "), e =
  c.GetVariable("$version ")
     }catch (f) {} if (!e)
 try {
  c = new ActiveXObject(d + ".6 "), e =
  "WIN 6,0,21,0 ",
c.AllowScriptAccess = "always ",
....
```

4. Experiments and Results

In this section, we present the preliminary results of our introduced web-browser forensic analysis method. Table 2 demonestrates the categorization for the 55 analyzed URLs.

Table 2: Categorization for the analyzed URLs

Description	Number of URLs
Compromised site leads to Angler Exploit Kit	15
Directs to Exploits	14
Compromised site leads to Exploit Kit	10
Mass	12
Script.Exploit	2
Exploit Kit	1
Trojan	1

Table 3: Distribution for the downloaded resources

Category	Number of files
JavaScript	249
PHP	42
Images	124
CSS	15

To generate the required trace files we accessed each of the 55 URLs separately using a virtual machine. The generated trace files were then passed to our developed analyzer. Figure2 demonstrate the output from the analyzer, it shows the number of occurance for some searched event like:

1) Vulnerability checking using ActiveXObject.
2) Vulnerability checking using Shockwave.
3) Downloading resources by assigning the src attribute.
4) Downloading resources using the iframe tag.
5) Created cookie files.
6) Encoding
7) Browser fingerprinting
8) Number of URL redirection

The x-axis represents the analyzed URLs and the y-axis represents the number of occurance for the searched events. One notable observation that most of the forensically analyzed URLs are profiling the users' behavior and fingerprinting their browsers. URL redirection, downloading external resources and creating cookie files are the most common events which are forensic evidences and can be used in our post mortem analysis.

After beautifying the extracted code we get a closer look for example into the type of downloaded files and the way used to download it. Table3 demonstrates the distribution for the founded files. A visualization for the file distribution is shown in figure 3

Our experiment showed that we can get a detailed trace file for any executed malicious JavaScript and if this JavaScript file tries to execute a php file or load a malicious ad for example, by creating an iframe and setting the source attribute to that page we can find traces for that file as shown in the code below.

```
iframe.src = iframe_url;
iframe.style.display = "none ";

document.write('<div id= "slide_up ">
<div id= "close_btn_noCookie ">X
```

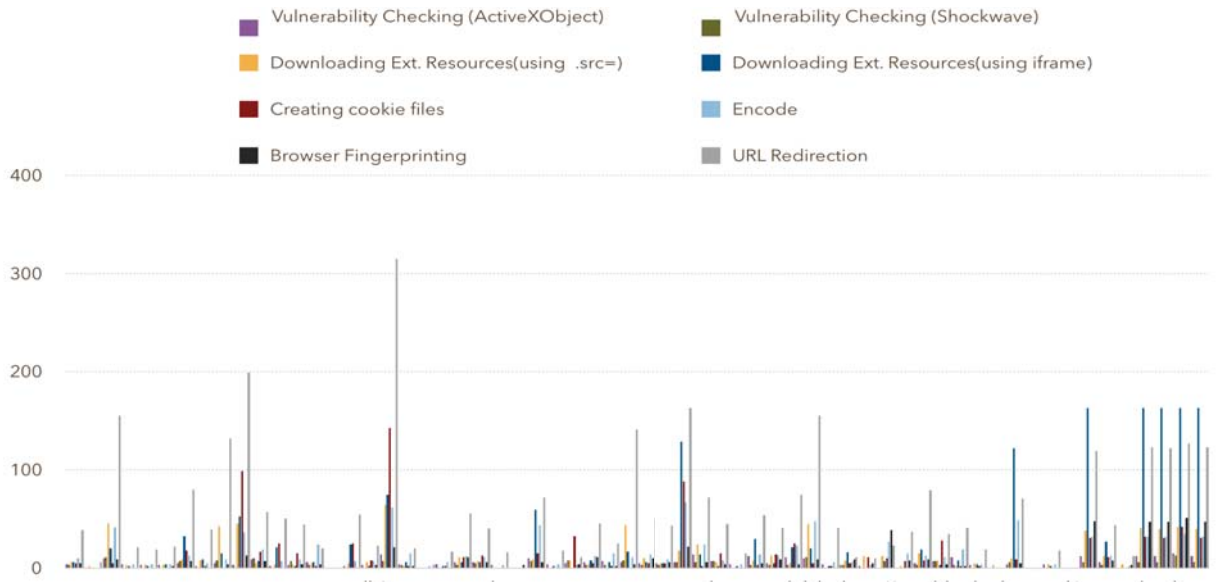


Fig. 2: The Distribution for the searched events

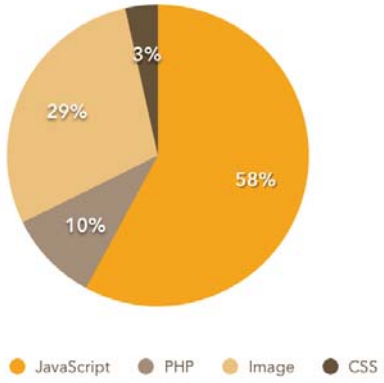


Fig. 3: Files Distribution

```
</div><iframe id= "su_frame " src=
"xxxpcash.imlive.c../releasese/...
```

We also use jsunpack-n[11] program, which emulates browser functionality to detect malicious code and we compare the list of JavaScript files analyzed by our Firefox extension with the files analyzed by it. We consider this comparison as a benchmark for our gathered data. For 36 URL, our extension has successfully detected 61% of the JavaScript files detected by Jsunpack-n and detected 24% of the JavaScript files for the rest of the analyzed URLs. It also detected and analyzed 54 JavaScript files which were not detected or mentioned by Jsunpack-n. Our explanation is that the developed extension only extract script files whose functions were loaded and executed in memory and those script files have no executed functions and were not loaded in memory while loading the page especially that most of these files are responsible for UI interactions like draggable.min.js, menu.min.js, mouse.min.js and so on.

[11] **Jsunpack-n:** is a command-line Javascript unpacker that has more or less the same features as the web version of Jsunpack

5. Conclusion & Future Work

Web-based attacks are gaining an increasing momentum and attention from cyber criminal and security researchers; alike, i.e. Drive-by-Download attack vector analysis, detection and prevention. In this paper, we introduce a forensic analysis method to examine web-browsers artifacts produced by accessing malicious URLs. A Proof-of-Concept Firefox Browser extension is developed to enable getting detailed information about the attack, techniques used to evade the detection tools, downloaded malicious resources and executed malicious code. Digital evidence trace file is output for each examined URL that contains a set of volatile and non-volatile forensic evidences that would assist a forensic analyst in the investigation. Our approach gives a closer look at the real code executed from the client side. In the future work, the introduced implementation will be extended to the different browsers, i.e. Google Chrome and IE.

References

[1] Zhang, J., Seifert, C., Stokes, J.W. and Lee, W., 2011, March. Arrow: Generating signatures to detect drive-by downloads. In Proceedings of the 20th international conference on World wide web (pp. 187-196). ACM.

[2] Egele, M., Kirda, E. and Kruegel, C., 2009. Mitigating drive-by download attacks: Challenges and open problems. In iNetSec 2009âĂŞOpen Research Problems in Network Security (pp. 52-62). Springer Berlin Heidelberg.

[3] Ligh, M., Adair, S., Hartstein, B. and Richard, M., 2010. Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code. Wiley Publishing.

[4] Jayasinghe, G.K., Culpepper, J.S. and Bertok, P., 2014. Efficient and effective realtime prediction of drive-by download attacks. Journal of Network and Computer Applications, 38, pp.135-149.
[5] Kotov, V. and Massacci, F., 2013. Anatomy of exploit kits. In Engineering Secure Software and Systems (pp. 181-196). Springer Berlin Heidelberg.
[6] Ohana, D.J. and Shashidhar, N., 2013. Do private and portable web browsers leave incriminating evidence?: a forensic analysis of residual artifacts from private and portable web browsing sessions. EURASIP Journal on Information Security, 2013(1), pp.1-13.
[7] Aggarwal, G., Bursztein, E., Jackson, C. and Boneh, D., 2010, August. An Analysis of Private Browsing Modes in Modern Browsers. In USENIX Security Symposium (pp. 79-94).
[8] Sonntag, M., Automating Web History Analysis. na.
[9] Van Overveldt, T., Kruegel, C. and Vigna, G., 2012. FlashDetect: ActionScript 3 malware detection. In Research in Attacks, Intrusions, and Defenses (pp. 274-293). Springer Berlin Heidelberg.
[10] Laskov, P. and Åărndiäğ, N., 2011, December. Static detection of malicious JavaScript-bearing PDF documents. In Proceedings of the 27th Annual Computer Security Applications Conference (pp. 373-382). ACM.
[11] Cova, M., Kruegel, C. and Vigna, G., 2010, April. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In Proceedings of the 19th international conference on World wide web (pp. 281-290). ACM.
[12] Curtsinger, C., Livshits, B., Zorn, B.G. and Seifert, C., 2011, August. ZOZZLE: Fast and Precise In-Browser JavaScript Malware Detection. In USENIX Security Symposium (pp. 33-48).
[13] Canali, D., Cova, M., Vigna, G. and Kruegel, C., 2011, March. Prophiler: a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th international conference on World wide web (pp. 197-206). ACM.
[14] Ratanaworabhan, P., Livshits, V.B. and Zorn, B.G., 2009, August. NOZZLE: A Defense Against Heap-spraying Code Injection Attacks. In USENIX Security Symposium (pp. 169-186).
[15] Rieck, K., Krueger, T. and Dewald, A., 2010, December. Cujo: efficient detection and prevention of drive-by-download attacks. In Proceedings of the 26th Annual Computer Security Applications Conference (pp. 31-39). ACM.
[16] Xing, X., Meng, W., Lee, B., Weinsberg, U., Sheth, A., Perdisci, R. and Lee, W., 2015, May. Understanding Malvertising Through Ad-Injecting Browser Extensions. In Proceedings of the 24th International Conference on World Wide Web (pp. 1286-1295). International World Wide Web Conferences Steering Committee.
[17] Choi, J.H., Lee, K.G., Park, J., Lee, C. and Lee, S., 2012. Analysis framework to detect artifacts of portable web browser. In Information Technology Convergence, Secure and Trust Computing, and Data Management (pp. 207-214). Springer Netherlands.
[18] Oh, J., Lee, S. and Lee, S., 2011. Advanced evidence collection and analysis of web browser activity. digital investigation, 8, pp.S62-S70.
[19] Kozushko, H., 2003. Digital evidence. online], http://infohost. nmt. edu/ sfs/Students/HarleyKozushko/Papers/DigitalEvidencePaper. pdf.
[20] De Maio, G., Kapravelos, A., Shoshitaishvili, Y., Kruegel, C. and Vigna, G., 2014. Pexy: The other side of exploit kits. In Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 132-151). Springer International Publishing.

Digital Forensic Analysis of SIM Cards

Mohamed T. Abdelazim, Nashwa AbdelBaki, Ahmed F. Shosha
Information Security Department, CIT School, Nile University, Cairo, Egypt

Abstract— *Smart cards are fundamental technology in modern life. It is embedded in numerous devices such as GPS devices, ATM cards, Mobile SIM cards and many others. Mobile devices became the evolution of technology. It becomes smaller, faster and supports large storage capabilities. Digital forensics of mobile devices that maybe found in crime scene is becoming inevitable. The purpose of this research is to address the SIM cards digital forensics analysis. It presents sound forensic methodology and process of SIM cards forensic examination. In particular, the main aim of the research is to answer the following research questions: (1) what forensic evidence could be extracted from a SIM card and (2) what are limitations that may hinder a forensic analysis?*

Keywords: SIM Cards; Smart Cards; Mobile Digital Forensics; SIM Card Forensics;

1. Introduction

Mobile forensics has become a rapidly important forensic domain, since mobile phones have become a common source of digital evidence. Nowadays, smart phones, tablets, and smart cards contains a large information storage with increasing capabilities [1] which provides an extensive source of information that could leverage forensic analysis and investigation.

Unfortunately, it's a common knowledge that criminals don't prefer using smart phones, because it can easily be tracked using different methods. As such, they may prefer to use prepaid phones known as "burner phone" or simple non-smart phones. Although, recent reports that Law Enforcement are confronted with criminals and terrorists that are increasingly using different smart phone models and Law Enforcement are facing difficulties decrypting digital evidence residual in those phone without the phone manufacturing support. This doesn't mean that forensic analysis of non-smart phone is an obsolete domain. In fact, SIM cards forensic is a fundamental activity in any mobile phone forensics, and law Enforcement are regularly in a situation where it is required to extract basic information from a phone SIM card.

The existing techniques, methods, software (including commercial or open source) in mobile phone forensics are focusing on the analysis of smart phones. For example, extracting artifacts of applications installed on the mobile operating system, i.e. Android or IOS; however very limited research addressing the forensic analysis of Mobile phone hardware or SIM card. In particular, SIM card forensics is disclosed forensic area, and currently limited to a set of commercial applications that could extract limited forensic artifacts. SIM cards are the gateway between the mobile phones and the network, they store information that could be used for reconstructing user's activities, such as phone book, incoming and/or outgoing calls, sent and/or received messages, message's timestamps, when the user power off or on his/her device, etc.

The main aim of the proposed research is to address the digital forensics aspects of SIM cards and providing open source library to support extracting sound forensic evidence from a SIM card. More importantly, the research attempt to answer the following questions: (1) what information could be stored in SIM cards, (2) what can be extracted and/or recovered to assist a forensic investigation, (3) and limitation of SIM cards forensic analysis.

The paper begins with a brief introduction about Smart phone forensics, Smart cards and SIM cards, in section Two. Section Three describes the forensic analysis methodology of SIM cards. Section Four addresses the methods and protocols that would enable communication with SIM cards and the SIM card's file system structure. In section Five, the method to forensically acquire SIM cards is presented, and section Six presents the SMS forensic analysis methods. In section Seven, a Proof-of-Concept implementation for extraction of forensic evidence from SIM cards is introduced, and sample experiments on different SIM cards models are presented. Finally, the last section presents the research conclusion, limitations and draws the future work. In summary, the research contributions are the following:

- Review of SIM card structure and required development environment.
- Review of SIM card file system, identifiers types and structure.
- Listing of communication methods with a SIM card.
- Forensic method that allows extracting of digital evidence from a SIM card.
- A proof of concept implementation for the proposed forensic methodology.
- Forensic timeline analysis recovered messages from SIM card.

2. Overview and Backgrounds

SIM cards analysis is not a new research, however, it's a disclosed research and very limited information about

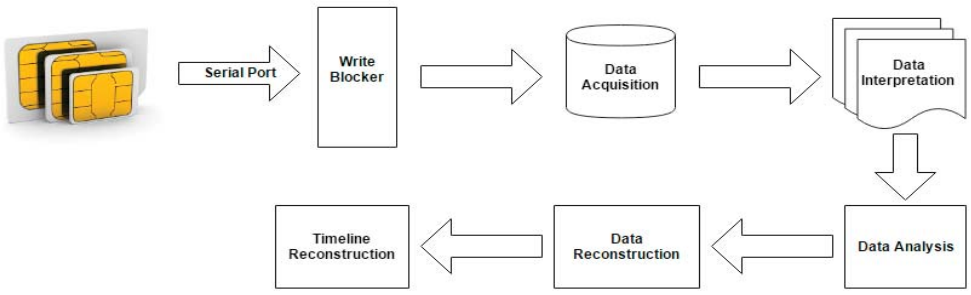


Fig. 1: SIM Card Forensic Analysis Process Flow

it are publicly available. The earliest SIM cards forensic analysis research is presented in [2] where the SIM file system structure and data residual in a SIM card were briefly discussed. Other published research proposed set of forensics tools such as [3], [4] and `SIMBruch` presented in [5]. However, those tools only focused on extracting the basic information from a SIM card and didn't provide clear description about the forensic techniques used to extract an evidence from the SIM card.

Moreover, the current available public forensic tools for SIM card analysis are not updated, and some of those tools are depending on the presence of specific currently unsupported hardware such as SIM readers compliant to PC/SC specification [6]. Because of those limitations, this research aims to provide an updated implementation that support extraction of digital evidence from modern SIM card models. Also, the proposed implementation is hardware agnostic and it communicates with a SIM card using the standard communication protocol. This will allows forensic researchers and/or Law Enforcement to use and extend the proposed research without having relied on a specific SIM hardware readers.

2.1 Smart Cards

Smart Cards are considered standalone embedded systems, where SIM cards are a subset of smart cards. Smart card's architecture are consists of EEPROM, FLASH, and processor that could be used to store information or execute operating system [7]. In addition, the design of smart cards allows user applications to be developed and installed on the cards using Java card language [8].

In general, smart cards have three different types: Contact Cards, Contactless Cards, and Hybrid Cards [9].

- Contact Cards: This type requires direct contact with another hardware to exchange information and to read the information stored inside the card, such as credit cards, debit cards or loyalty cards [10].
- Contactless Cards: This type does not require any direct contact to any hardware. These cards communicate using RF protocol, wireless, or Bluetooth such as employee ID cards [11].
- Hybrid Cards: This type of cards is able to use both methods of communication either by direct contact or not, such as smart RFIDs, some credit cards models, and fingerprint cards [12], [13].

2.2 SIM Cards

SIM cards are contact cards. It is present in every mobile phone, and originally developed for secure communication and GPS devices. Unfortunately, SIM cards are becoming one of the main attack vectors that enable a remote access control over the device and to turn a mobile phone to a listening device and record the user's conversations and/or to steal personal information [5], [2].

SIM cards could be accessed using the operating device or using external reader. It is considered very important in the mobile forensics process. It can be used to hide information or lock the mobile, which will make the mobile investigation harder, and can delete some information from the device at removal.

2.3 Development Environment

SIM cards development environment depends on Java technology known as Java card. Java card is structured to three main components [14]:

- Java card virtual machine: used for application installation and defining the features and services.
- Java card run-time: which handle context switching between applications.
- Java card API: which is a set of classes and interfaces to allow the application to use card services.

Java cards inherit the security from Java programing, this environment allows multi-application to be installed and run, which allow developing application or even malwares [15].

3. Methodology

In this section, the forensic analysis methodology applied in this research is presented in figure 1. It begins with data

acquisition where the forensic data is extracted from the SIM using serial communication. In this stage a forensic write blocker is used to prevent any accidental attempt to tamper the SIM card file system or files' content and to ensure a sound forensic analysis.

In the second stage, the acquired SIM card data is interpreted. This includes extracting the SIM file system and interpreting the file system structure for analysis.

In the third stage (aka. data analysis stage), forensic evidences were identified and highlighted. This includes for example, identifying files that contain the user phone book or incoming calls list, etc.

The final stage represent the forensic event reconstruction, where digital evidences are presented in timeline representation to assist forensic investigator to develop an understanding of the identified evidence context, reconstruct user behavior on the phone, and recover (or at least identify) evidence that may have deleted from the SIM card.

4. SIM Communication

Smart cards use two types of communication. (1) Application Protocol Data Unit command (APDU) that executes commands on binary level. (2) `AT` command, that is used to communicate with the SIM on the application level using well defined interfaces.

GSM communication protocol provides a set of well defined APIs to extract information from SIM cards. In particular, there are two possible methods for data extraction using (1) `AT` command or (2) accessing a memory address on a binary format. The back-draw with the later method is that not all memory addresses are available to be accessed, as SIM cards specification provides a set of restrictions and limitations on the executed command [16].

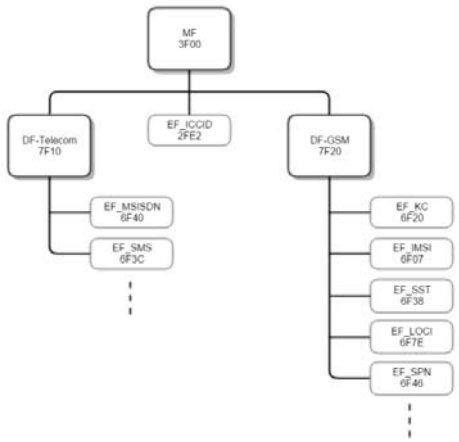


Fig. 2: SIM File System Structure

4.1 SIM File Identifier

SIM card file system consists of set of files in binary format. Figure 2 descries sample of SIM file system tree where there are three types of files identifiers [17]:

- Main Files (MF) that resides at memory address `0x3F00`. MF are considered the root directory for SIM file system.
- Dedicated Files (DF): it represents the application level for each service provided by SIM such as directories.
- Elementary File (EF): it holds the actual information which needs to be extracted for digital forensics.

Each type of file identifiers is consisted of header and body. All file identifiers have headers which hold the file size, permission access which could be summarized as five permissions [4], data storage type, and file ID. However only EF file has body which contains the actual data. Elementary files (EF) has mainly two types of data storage (1) Plain data where the date is stored under the EF file, which can be accessed by EF ID and file size; (2) List (records) where the data is stored as list (array), which can be accessed by record ID and record size.

5. Data Acquisition

SIM card forensic acquisition can be achieved by two methods (1) physical or (2) logical acquisition. In physical acquisition a bit-by-bit forensic image of SIM memory is acquired for analysis. In the logical acquisition, the SIM data structure, i.e. file system structure and content is extracted and parsed for analysis. In this research, logical forensic acquisition method is used. This will be enabled by extracting forensic data from a SIM card in native format and interprets it for forensic analysis. In the following we will present information that could be extracted from a SIM card:

- Integrated Circuit Card Identifier (ICCID): Each SIM is identified by a unique integrated circuit card identifier internationally. ICCID length is defined by GSM as 10 octets (20 digits). The address of EF file containing this information is `0x2FE2`.
- International Mobile Subscriber Identity (IMSI): Each SIM has a unique identifier on each individual network provider. IMSI size is 15 digits and its EF file located at address `0x6F07`.
- SIM operation mode: there are three operation modes which are, data, fax class 1, and fax class 2.
  Most SIM cards are operating in Data operation mode. Using `AT+FCLASS?` command we can retrieve the SIM operation mode.
- Service Provider Name (SPN): SP is the name of the service provider where the SIM is registered. The size is defined by GSM standard as 17 bytes. The EF `EF_SPN` file holds this information located at address `0x6F46`.
- GSM Ciphering Key(Kc): Kc is one of two keys used to authenticate with GSM networks. This key is used

along with Ki to perform the handshaking protocol. The key can be recovered from address 0x6F7E.

- SIM Service Table: This file EF_SST indicates which services are allocated or activated. Each service is constructed with two bits, the first bit indicates if the service allocated or not; "1" means allocated, "0" means not allocated. The second bit indicates if the service activated or not; "1" means activated, "0" means deactivated [16].
- Short Message Sent (SMS): To read SMS saved within the SIM we can use API AT+CMGL provided by GSM protocol. This command will return all the SMS within the SIM in hex format. SMS is considered point-to-point communication protocol. There are two types of SMS, first is data SMS which contains normal text, second data download SMS, which contains instructions or executable code to be executed on the SIM. After executing AT command to read the stored SMS on the SIM card, the result execution status code could be 90 00, 91 xx or 7F xx. In case of 90 00 or 91 xx then this means that the current record is data SMS and the user will get notification. In case of 7F xx then the current record is SIM data download and the user will not get a notification for this SMS. SIM data download messages are sent directly to be executed by SIM card.

Other Information like PIN key state, and phone operation mode also could be extracted with a specific API which we will demonstrate later.

6. SMS Forensics

Short Message Service (SMS) is essential artifacts in mobile phone forensics, by developing a solid knowledge about the SMS data structure and how SMS is stored in the SIM card, forensic analyst can use it to reconstruct a human activity in a mobile phone subject for forensic investigation. Each SIM card contains limited number of memory locations, commonly it uses 40 locations with length size of 176 bytes each to store incoming or outgoing SMS. Each location structured in two parts: (1) records the status that is defined by its first byte of the record, indicating if the record is used or free. (2) Message's data; which holds the rest of the record with maximum size of 175 bytes.

Each SMS location initializes the first byte with "00" and the rest of the record with "FF" indicating unused location, at the point of incoming or outgoing new message, the message is stored in the first available free location. At the time of message deletion, the assigned record is reset to its initial value removing any trace of the original message.

By knowing that each message is marked by a timestamp and its stored location index, messages timeline construction can be achieved by providing time frame for received, sent, deleted, or overwritten messages which will be demonstrated later on[18].

7. SIM Card Forensics Proof-of-Concept

In this section, PoC to extract artifacts from a SIM card for forensic analysis purposes is developed. It allows extracting the basic information from the SIM subject for analysis, given into consideration to avoid any usage of write commands. This will ensure a sound forensic analysis and will avoid tampering with a SIM card as digital evidence. The PoC is developed based on the GSM communication protocol standard.

The developed software communicates with SIM card using serial communication. API interfaces is developed that can be used to extract certain pieces of information or for full data extraction using python programming language. It initiates a communication to the SIM card that allows executing a stream of AT commands on the SIM. By using AT commands we can avoid the need for having reader compatible with PC/SC specifications even using USB modem can achieve the same results.

The developed software extracts several valuable forensic artifacts, such as: saved contact lists, SMSs, the card identifier, service provider, part of the GSM location information, card service table, and many other information[19].

To ensure correctness of the PoC, a number of experiments are conducted using different SIM cards models that operates under different service providers, including one SIM card that is un-allocated to any service provider[1]. The experiments are presented and results are explained, as follows:

7.1 Case 1.a:

In this experiment, we used operational SIM card working for several years in attempt to identify user activities and extract stored SMS. As such, we been able to extract raw date for valuable forensic information and interpret it as follows:

(1) Extracting the EF_LOCI that holds Temporary Mobile Subscriber Identity (TMSI), (2) Location Area Information (LAI), (3) TMSI TIME, and (4) Location update status. The EF_LOCI file size is "11" bytes. Using the AT commands to read the EF_LOCI address AT+CRSM=176,INT(0X6F7E),0,0,11, it resulted the following:

```
CRSM:144,0,"5CD749D162F2201CDE0000"
OK
```

Another successful attempt to extract SIM service table using AT+CRSM=176,INT(0x6F38),0,0,14 and SIM card response with the following:

```
CRSM:144,0,"9EEF1F9CFF3E000000FFFFFFFFFF"
OK
```

In addition, we were able to extract the contact list saved on SIM card using AT+CPBR=<param1>, <param2>

[1]service providers have been eradicated

command, which reads the phone book starting from <param1> to <param2>. SIM cards can hold about 255 record and using AT+CPBR=? we can get phone book properties, such as: how many records can be stored on the SIM, the maximum number of digits for each number, and the maximum number of character for each text. Below, a sample of the results:

```
1) Name: "Luck", Number: "01004677080"
2) Name: "Cris", Number: "01006059135"
3) Name: "Jaky", Number: "027180749"
```

Another extremely valuable forensic information to extract is "pin key status" for PIN1 and PIN2. This is enabled using AT+CPIN? that can be used to return the state of PIN1 key and AT+CPIN2? and get the state of PIN2. SIM operation mode can be extract using the same method. Below, a sample of the results:

```
Pin 1: READY
Pin 2: Not Found!
Operation Mode: Data
```

Another information that can be extracted is the SIM CCID and IMSI. This is enabled using AT+CRSM=176,int(0x2FE2),0,0,10 for CCID and AT+CIMI for IMSI, as follows:

```
CCID: 89**************385F
IMSI: 602**********385
```

For SMS forensic extraction, saved messages can be extracted using the following AT+CMGL=4 command which will list all the messages including message header information and message content, as follows:

```
Message ID: 4
Message State: received read messages
Length: 30
Content: 07910211970028646 40ED0457A7A1E66
87E900085101012114718 00A05000378050506290
029
```

SIM service provider name can also be extracted using the below code: AT+CRSM=176,int(0x6F46),0,0,17.

```
Service provider name: *****
```

Finally, SIM service table is extracted using the below code: AT+CRSM=176,int(0x6F38),0,0,17.

```
SIM Service Table(SST): 9EEF1F9CFF3E000000
FFFFFFFFFFFFFFFF
```

7.2 Case 1.b:

In this experiment, a brand new SIM card is forensically analysis to determine what minimum information can be extracted for the forensic purposes. The SIM does not contain any phone book records. Thus, we been able to extract limited raw data for some information and interpret it, as below: The SIM CCID, IMSI number, the stored SMS, operation mode, and PIN status were successfully extracted. However, SIM service table couldn't be extracted. Below a sample of the resulted data:

```
Message ID: 9
Message State: stored unsent messages
Length: 22
Content: 07910221020020F0110000000000A8104
9503B3D9F83F2EFBA0B04028140

SIM CCID: 892*************213
SIM IMSI: 602********941

Service provider name: ******

Operation Mode: Data
```

7.3 Case 2:

In this experiment, we used a blank card that was not activated with a service provider. Following, fabricated data, such as some phone book information, SMS information have been inserted and were forensically extracted. Since, the SIM was not registered to any service provider, the service provider name was set to the default value "FF", and fabricated data was correctly extracted. Below, a sample of the extracted data:

```
SIM CCID: 898*************0529
SIM IMSI: 460**********34

1) Name: "test name ", Number: "545465465"
2) Name: "SOS", Number: "112"

Message ID: 1
Message State: received read messages
Length: 25
Content: 0004068121856400005090723151808A0
AF372195407CDDF6F37

Operation Mode: Data

Service provider name: None
```

Based on the above experiments, one essential finding can be determined, in which is having non-registered SIM card does not refute the potential of having forensic artifacts, and necessity of investigating blank SIM card as it may contain valuable information.

7.4 Case 3.a:

In this experiment, a new SIM card that is locked with PIN key is analyzed. In this case, if the mobile phone restarted or powered off, user will not be access the SIM until a PIN key is entered. Below, a sample of the extracted information:

```
PIN Status: SIM PIN

SIM CCID: 89*************052
SIM IMSI: Not Found!

SIM Service Table(SST): Not Found!

Service provider name: ****

Operation Mode: Data
```

Based on the above output, we can conclude that essential information such as: IMSI, SST, phone book, and SMS are protected when PIN key is set. Thus, to allow extracting forensic artifacts from a SIM card, it's essential to disable the set PIN key, otherwise forensic acquisition using this method will not be successful.

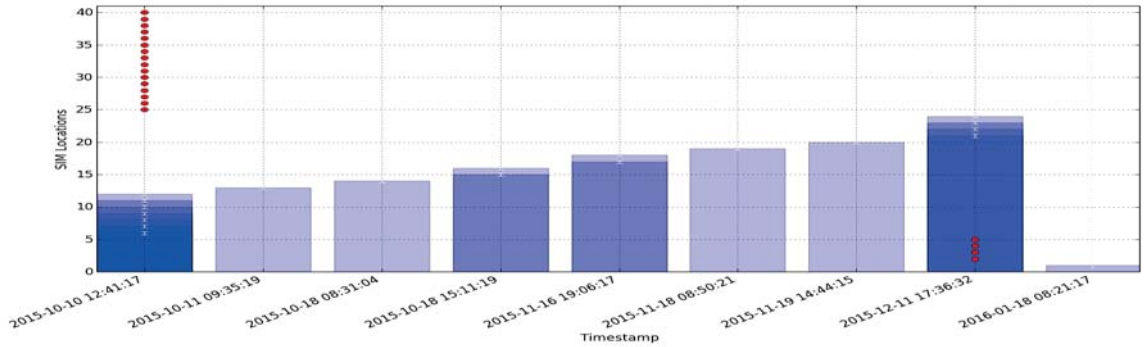


Fig. 3: Forensics Reconstruction of SMS Messages

7.5 Case 3.b:

To ensure Case 3.a finding and conclusion, the same SIM card with PIN key disabled is forensically examined, and below is sample of the results:

```
SIM CCID: 894***********052
SIM IMSI: 262*********205

1) Name: "M*****x", Number: "********"
80) Name: "Ta*******n", Number: "****"
81) Name: "Ko******er", Number: "****"

Message ID: 1
Message State: received read messages
Length: 159
Content: 07919471071600000040C9194710707520
000125101706085908 0A0D7349BBD7EB7DB6537485
C4E83C4EC70DD452E87406276B...

Service provider name: ****

Operation Mode: Data

PIN Status: READY

SIM Service Table (SST): 9E3B140C27FE5DFFF
FFFFFFFF
```

We can conclude that if SIM PIN key was disabled, a forensic analyst will be enabled to extract SIM content for analysis such as: IMSI, contact list and SIM service table.

7.6 Case 4:

In this experiment, we are using the SMS forensic information we have previously examined to construct SIM activities timeline. The experiment shows that each new SMS message is stored at the first empty location, where each SMS has its timestamp which was originated with the message. Using these two findings, we can reconstruct some of the SIM activities and provide timeline of the messages received. The experiment was conducted over operational SIM cards and different mobile phones (burner and smart phones) with few SMSs saved in an attempt to build time frame of saved and/or deleted messages.

1) SMS in Smart Phones: Using operational SIM card placed in a smart phone, it was possible to extract messages received from entities such as: service providers and Banks, some advertisers, etc. However, user sent messages, were saved on the smart phone internal memory, not in the SIM card. Based on SMS timestamp, figure 3 displays received messages by its timestamp and the SIM storage index.
2) SMS in Burner/Non-Smart Phones: Using a burner or non-smart phone with operational SIM card, it is possible to extract, not only, the service provider messages, but also, the user messages received from other users. This emphasis the importance of SIM card forensic, in case of investigating non-smart phones; As in this case SIM cards may only be the viable digital evidence source.

Figure 3 explains PoC of the timeline representation for the received messages where the experiment was conducted using operational SIM card with "40" locations available and few messages saved. The graph shows the number of received messages on a particular day represented by over layered bars i.e. at timestamp `2015-10-10 12:41:17` which shows that seven messages have been received and stored at locations "6" through "12".

Also the graph represents the possibility of the message deletion at a certain timestamp which is represented by "red circle" i.e. at timestamp `2015-12-11 17:36:32` locations "2" through "5". The rest of the empty locations "26" through "40" at timestamp `2015-10-10 12:41:17` have one of two possible meanings, (1) These locations have never been used; (2) These locations have been freed.

7.7 Case 5:

In this experiment, we are extracting the headers of the files by scanning the SIM address space starting from `0000` to `FFFF` using the `GET RESPONSE` command with instruction ID `192`:

```
AT+CRSM=192,INT(File_ID),0,0
```

Below, a sample of extracted raw file header:

```
File ID: 6f38, File Header: 0000000A6F3804001BFF
```

```
FF01020000
File Size: 10, Raw Data: FF3FFF0F0300F003000C

File ID: 6f4a, File Header: 000000416F4A040011F
FFF0102010D
File Size: 65
Record ID: 1, Record Size: 13 ,Raw Data: FFFFFFF
FFFFFFFFFFFFFFFFFF
Record ID: 2, Record Size: 13 ,Raw Data: FFFFFFF
FFFFFFFFFFFFFFFFFF
```

Table 1 shows some of the byte representation for 15 bytes header. It differentiates between EF, DF and/or MF files and identify the data structure for EF files which allows extracting the raw data.

Table 1: File Header byte mapping

Byte Number	Description
1-2	Null bytes
3-4	File Size
5-6	File ID
14	File data structure which differentiate DF and EF
15	Record Size (length)

8. Findings and Limitations

In this paper, the method and protocol to communicate with SIM card for forensic analysis purposes is explained. This includes, what information could be extracted, and how to extract it. A proof-of-Concept implementation was also introduced to automate the investigation and assist forensic analysts. However, the developed software is confronted with some limitations; Most notably, SIM cards that are protected with PIN key. If it is active, extracting information form the SIM card for forensics purposes will not be available using this method. In addition, most of SIM cards have only three attempts to enter the PIN key, which hinder the method of brute forcing the PIN key.

Although SMS messages can be extracted from the SIM card with the user's messages, however deleted messages can't be recovered using the proposed approach in this research, which is an essential requirement in forensics analysis. Another limitation to the current method, it is not currently possible to determine or confirm user activities after the last stored SMS message, where there could be another messages received and deleted, the same scenario also applied for the first recorded message.

9. Conclusion

In this research, SIM cards, SIM file system structure, and the communication methods that would enable forensic data extraction were explained. The research demonstrated how a forensic analyst could extract vital forensic artifacts from a SIM card, such as: SMS, phone book and card CCID. Limitation and constrains also were presented.

SMS are core artifacts in forensics investigation. Using SMS timestamps and stored indexes, forensic analyst could construct a timeline for the SIM activities. In particular, it is possible to identify if message(s) have been deleted or overwritten. Also, the content of saved SMSs on the SIM can be extracted.

The current ongoing research is to extend SMS forensics to include messages sent from the mobile known as "USSD codes" or the commands sent to the mobile known as "push messages", and investigating the file identifier header is also essential which may allow bypassing PIN through changing the file permissions in the header. Also, it may shed the light on methods that might be used as an attack vector to install malware on a smart phone.

References

[1] I. M. Baggili, R. Mislan, and M. Rogers, "Mobile phone forensics tool testing: A database driven approach," *International Journal of Digital Evidence*, vol. 6, no. 2, pp. 168–178, 2007.

[2] W. Jansen and R. Ayers, "Forensic software tools for cell phone subscriber identity modules," in *Proceedings of the Conference on Digital Forensics, Security and Law*, 2006, pp. 93–106.

[3] R. Thakur, K. Chourasia, and B. Singh, "Cellular phone forensics," *International Journal of Scientific and Research Publications*, vol. 2, no. 8, 2012.

[4] W. Jansen, A. Delaitre, *et al.*, "Reference material for assessing forensic sim tools," in *Security Technology, 2007 41st Annual IEEE International Carnahan Conference on*. IEEE, 2007, pp. 227–234.

[5] F. Casadei, A. Savoldi, and P. Gubian, "Forensics and sim cards: an overview," *International Journal of Digital Evidence*, vol. 5, no. 1, pp. 1–21, 2006.

[6] Osmocom. Osmocom simtrace. [Online]. Available: http://bb.osmocom.org/trac/wiki/SIMtrace

[7] H. Guo, "Smart cards and their operating systems."

[8] "Digital cellular telecommunications system (phase 2+); subscriber identity module application programming interface (sim api) for java card," European Telecommunications Standards Institute, France, Standard, Mar. 2003.

[9] W. Rankl and W. Effing, *Smart card handbook*. John Wiley & Sons, 2010.

[10] K. Markantonakis *et al.*, *Smart cards, tokens, security and applications*. Springer Science & Business Media, 2007.

[11] A. Juels, "Rfid security and privacy: A research survey," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 381–394, 2006.

[12] V. Meireles and P. Benato, "Double-sided electronic module for hybrid smart card," Dec. 23 2005, uS Patent App. 11/315,273.

[13] G. Kayanakis, "Contactless or hybrid contact-contactless smart card designed to limit the risks of fraud," May 21 2002, uS Patent 6,390,375.

[14] O. Inc. Java card technology. [Online]. Available: http://www.oracle.com/technetwork/java/embedded/javacard/overview/getstarted-1970079.html

[15] I. Sun Microsystems, *Java Card Applet DeveloperâĂŹs Guide*, 1998.

[16] "Digital cellular telecommunications system (phase 2+); specification of the subscriber identity module - mobile equipment (sim-me) interface, 3gpp ts 11.11," European Telecommunications Standards Institute, France, Standard, Mar. 2005.

[17] "Universal mobile telecommunications system (umts); characteristics of the usim application," European Telecommunications Standards Institute, France, Standard, Oct. 2005.

[18] "Digital cellular telecommunications system (phase 2+); technical realization of the short message service (sms) point-to-point (pp)," European Telecommunications Standards Institute, France, Standard, Dec. 2001.

[19] M. T. AbdelAzim. Simanalyzer tool. [Online]. Available: https://github.com/Tabaz/SIMAnalyzer

EasyAuth – Implementation of a Multi-Factor Authentication Scheme based on Sound, Fingerprint and One Time Passwords (OTP)

Levent Ertaul, Ishita Thanki
CSU East Bay, Hayward, CA, USA.
levent.ertaul@csueastbay.edu, ithanki@horizon.csueastbay.edu

***Abstract*—In this research paper, a multi-factor authentication scheme is facilitated in the form of an android application called EasyAuth that will improve the Login process of Twitter via three schemes: Voice/Sound Based Authentication, Fingerprint Authentication, and One-Time Password (OTP). There is a well-established and known 2-factor authentication scheme, however, the EasyAuth application is designed as a one-step advancement authentication scheme that integrates three types of authentication methods making use of multi-factor authentication with a random selection for those methods. The authentication process is explained step-by-step with help of code snippets for each of the schemes used in the EasyAuth application. This paper explains and gives detailed description on the Vigo library used in the voice authentication as well as SPass API used for fingerprint authentication. In addition to voice and fingerprint authentication, a randomly generated one-time password generation is used which is produced by a web-service. Finally, test results are shown for all three authentication systems used in the EasyAuth application which makes the login process much easier for end users by using the provided schemes that can be randomized further to improve security.**

I. Introduction

Mobile users are nowadays highly frustrated because they need to remember distinct passwords for different websites as well as for their Android applications that include a variety of requirements for setting a password. For example, common requirements include minimum length or combination of: upper case letters, lower case letters, special characters and numbers. Users also need to remember those passwords which can be an irritating process and every end user wants very high security but without bearing the stress and pain of creating and remembering lengthy passwords [1].

The first authentication scheme is voice based authentication using the Vigo library [2] that is provided by a company based in California named Voice Vault Inc. The Vigo library is a standardized approach to mobile voice biometrics that is based on the simple and overriding idea that mobile use cases are inherently the same: people are using the same devices, in the same environments to achieve the same goals. The Vigo approach means that building and deploying voice biometrics in a mobile app is as simple as possible and can be achieved in the shortest possible time and with a minimum of resources [23]. It is a mobile voice biometrics service which is working on two major guiding principles: Simplicity and Standardization. Vigo is hosted by Amazon Web Service (AWS) and there is no local infrastructure to deploy or maintain [3]; hence, the data is stored on the cloud and the Voice Vault manages and maintains it for you and assures you that it is secured.

The second type of authentication scheme used in EasyAuth is fingerprint authentication – the basic algorithm used here compares two fingerprints and upon finding the match, it grants access to the system. It is proved that every human has unique fingerprints and hence it is useful for authentication process [4]. EasyAuth uses Samsung Pass API to authenticate a user before providing access to a Twitter account. In order to make this feature work, the system requires a fingerprint sensor to digitally capture the image of users' fingerprints which is called "live scan". This live scan is digitally processed to make a biometric template out of it, which is stored and used to perform matching. There are various technologies used in the sensors which are piezoelectric, piezo resistive, ultrasonic etc. are used in the sensors.

To add another security layer to the above schemes, the idea of adding OTP is explored. OTP works like a token which changes every time or on periodic time intervals [5]. This is used to add an additional layer of security as it has been used in other 2-way authentication schemes and has been successful. A huge example of this scheme is Google Authenticator which provides users with this functionality but has some limitations too for third party applications [6]. OTPs have been proved better than a user selected password since passwords selected by users are relatively weak and can be guessed or cracked easily; contrary OTP's are not easily guessed since they change every 30 or 60 seconds.

In EasyAuth, a special provision is made so whenever users select from the above three schemes; a random scheme will be shown to them for authentication which is totally random and very hard to guess too. This is the one twist introduced for EasyAuth for making the system significantly secured. Further, users will be able to choose the different kind of authentication which may be either one of the users' choice or they can select 2-way or multi-factor as per their security requirements. This makes EasyAuth best fit and simple to use as based on users need.

This mechanism will solve the password frustration problem of the users since every scheme adopted in EasyAuth is making use of users' voice/finger biometrics that they do not need to remember. Section II of this paper addresses the comparison between 2-factor and multi-factor authentication.

ISBN: 1-60132-445-6, CSREA Press ©

Section III will highlight the details of how the multi-factor authentication method was implemented in EasyAuth, and provide the system architecture for the same. Section IV will present the test results obtained during testing EasyAuth application. Section V gives the final conclusion.

II. 2-WAY AUTHENTICATION VS MULTI-FACTOR AUTHENTICATION

There are different types of technologies used to provide authentication. Some of them include: 2-way authentication (2FA) or 2-step verification or multi-factor (MFA), and 2 or more step authentication.

The 2-factor authentication is basically based on two things: something you have and something you know. If a system provides authentication that fulfills the above two things, then this can be called a two-factor authentication system. For instance, a user can have his username and password and user will be provided by a token, which is a random number and is independent of the system. Once he enters his credentials to login to system and if the token which user entered matches then only he will be allowed to login otherwise the authentication fails.

The 2-factor authentication is very unpopular because it involves extra steps that the user must complete in order to log into the system. Traditionally, hardware tokens like RSA Secure ID [7] and dongles [8] were used which complied with FIDO U2F [9] standard for universal two factor authentication. These methods may not be cost effective and need a physical token that is provided by a service provider to each user. Nowadays software tokens are being used instead of hardware tokens to make it easy and affordable cost wise. Google 2-Step Verification [10] is one such example, which allows users to enter verification codes.

But, these systems are giving a hard time to users because they need to wait for a code and sometimes it would take a long time to authenticate. One more flaw in Google Authenticator is that it is based on the local time of the system so if your time is incorrect then it will not authenticate you since the algorithm fails to work. Hence, it is important that your phone's time is accurate [10] [11]. Most of time 2-factor authenticator and multi-factor authentication are considered the same but they are not similar. One example is a corollary of mathematics which says "Every square is a rectangle but not every rectangle is a square". This means the criteria defining one object doesn't encompass the criteria for the second object in all situations [12]. Hence as both of these sounds too similar it is very common for many companies to market their products as having multi-factor authentication.

Basically, multi-factor authentication comprises of three criteria's – something you know, something you have, and something you are. These three factors are needed, the last factor "something you are" plays a very significant role since it will be very unique and independent from the system used. Further, there's no way an attacker can guess or have any idea of that authentication factor as it is based on something you are. For instance, the "something you are" factor can be a biometric of the user which may require them to give their voice biometric, fingerprint, retinal scanning, or facial recognition [13].

One can design a system where users' needs to go through multiple layers of security requiring them to enter their username and password followed by a verification code and finally authenticate with biometrics. But in spite of designing such systems, which make use of 2-factor authentication or multi-factor authentication, most of the users prefer password-only authentication for services where 2-factor authentication is not mandatory [14] [15]. This is probably due to the extra burden that 2 factor authentication causes to the user [16] [17].

The best example of multi-factor authentication currently is Microsoft Azure multi-factor authentication. They provide many different ways by which user can authenticate themselves and hence called multi-factor authentication. The Microsoft Azure has one additional fraud reporting feature in which a user gets a message asking to proceed for step-2 verification using OTP. At that time user can press a Fraud Report button, which will help in detecting frauds [18].

There are different vendors who provide MFA solutions like EMC RSA Authentication Manager which is part of its SecureID technology, Symantec Verisign VIP, CA Strong Authentication and Vasco Identikey Digipass [29]. There is one more alternative to this that is offered by LastPass [30] which gives 2-factor authentication. Also PCI will make use of MFA which has been proposed recently [31].

On further research, one can make up a point that it is impossible to use such methods which provide high security but are frustrating to users where users need to remember passwords and again enter some more information to secure their systems. One such system which makes use of multi-factor authentication, which gives users the freedom to authenticate themselves very easily as well as with their biometrics, could be one solution to reduce the password frustration and make the authentication process easy for users which is discussed in the next section.

III. EASYAUTH APPLICATION - IMPLEMENTATION

In order to provide ease to users to authenticate themselves we came up with an idea of building and designing an android application called "EasyAuth". The main objective behind creating this application was to provide users a very simple mechanism to make the login process easier. Twitter accounts are too much likely as well as relatively easy to hack. History shows that approximately 250,000 Twitter account passwords have been compromised by hackers [20] and Twitter suffered high-profile spate of hacks in 2013 [21].

To overcome this, Twitter implemented 2-factor authentication but it was not mandatory for users and thus asked users to create strong passwords. However, recent news shows that "Twitter is emailing users whose account security was compromised by a bug last week, exposing email addresses and phone numbers linked to "a small number of accounts." The company also said that fewer than 10,000 accounts were affected [22]. The above news clearly shows that Twitter accounts are very much targeted by hackers, and it would be difficult for twitter users to secure their accounts if the accounts get hacked even after implementation of 2-factor authentication by the company.

We figured out how to build such an application which uses authentication factor which is "something you are" and came up with an idea of providing enhancement to Twitter users by giving them an option of multi-factor authentication with random selection. The system will only be able to decide on the basis of selected options provided by users which will make the login process of Twitter very convenient and difficult for hackers to guess and crack too.

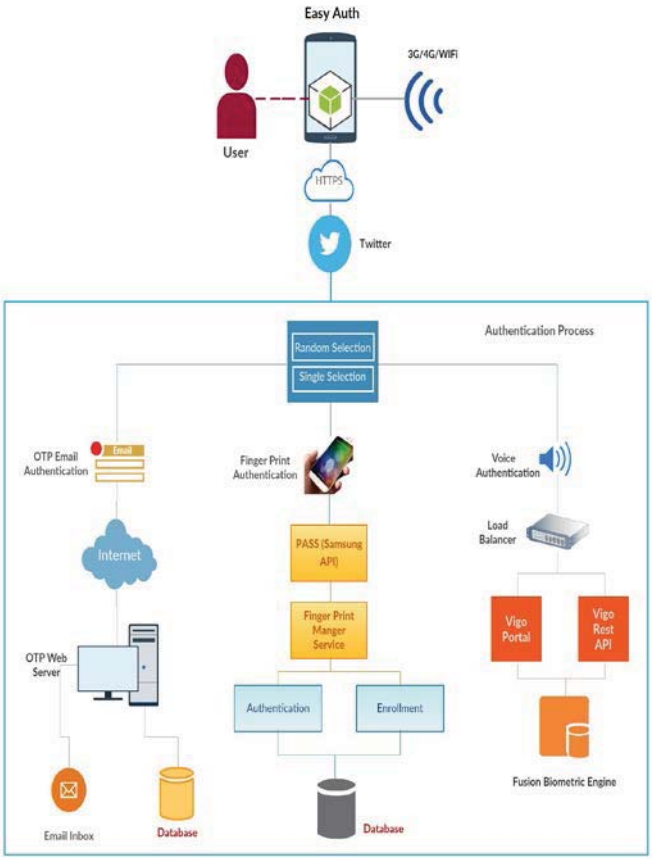


Fig. *1 System Architecture Diagram for EasyAuth Application*

Figure 1 above depicts the system's architecture of EasyAuth and gives an idea about how the functioning of the EasyAuth application and various components used to build this application. This includes various APIs used to integrate Twitter with the different types of authentication schemes provided by the application in Android OS. As shown in Figure 1, the user needs to login to Twitter.com via EasyAuth; we have used Twitter -4j API [18] to integrate Twitter login with EasyAuth which is done securely using https connection. The users need a smart phone and the internet to download and install EasyAuth.

Furthermore, users will login to Twitter using their Twitter ID and password which is a single type of authentication. In integrating Twitter login to EasyAuth, whenever user logs in Twitter – an Auth token is generated and sent back which is further used to retrieve user's information for user Tweets. Once they login, users will be provided three types of authentication schemes which acts here as multi-factor authentication. These includes authentication with voice, fingerprint, and OTP. We have given flexibility to users to select any of one, two, or all the three schemes based on their requirements as well as the level of security they need.

Once they select the authentication scheme, they will be asked for registration with the preferred scheme. For instance, if a user selects voice based authentication, then in the next phase, a user will be asked to register his/her voice. This will be done with the help of Vigo Library where a user will be asked to speak phrases composed of 4-digit codes, i.e., "1234". This process is repeated until the user's voice is successfully stored in the cloud server used by Vigo [23]. Next time when users logs in, they will be asked to speak a random 4-digit code and the spoken phrases will be matched with the voice samples stored in the Fusion Biometric Engine whenever users attempts to log in to Twitter.com using EasyAuth.

The second authentication scheme is Fingerprint Authentication which asks users to log in to Twitter using their fingerprint. We have used Samsung Pass API [24] to implement this in EasyAuth where users' needs to first register their fingerprints using the enrollment module, and later they will verify their fingerprint at the time of authentication by placing their finger on the hardware sensor which is provided by most recent Samsung Galaxy cell phones including Samsung Galaxy S6 Edge, Samsung Galaxy S7 etc. In this way, users will be able to use the inbuilt fingerprint sensor for authentication making this cost effective, fast and secure.

The third scheme is Time-Based One Time Passwords authentication scheme. To use this, we make use of a web-service which is called to send an OTP to the email of a user which he/she will enter when asked for by this type of authentication during the login to Twitter. The only reason we ask users to enter the email address again during login is due to security issue of Twitter's API which will not allow the storing of credentials for security reasons. Also, we believe that user credentials are highly confidential and must never be stored anywhere in the application.

In EasyAuth, there is a twist to enhance the security to its pinnacle because if a user selects all three schemes, next time whenever a user wish to login to Twitter, the system itself will give any one of the three schemes to provide authentication. The scheme chosen will be random, making it very hard for an attacker to guess what kind of authentication scheme will be asked by system. This is very significant and this is the most important aspect of our project.

The benefit of random selection is tremendously useful to users since they do not need to go through several steps in order to authenticate themselves. This will definitely avoid many problems which lead users to use multi-factor authentication schemes and help them login to their Twitter account easily without any issues of entering any extra information.

This project was implemented on a personal computer with a 2.5GHz Intel i7 processor, 8 GB of RAM, and running on a Windows 8.1 OS. Android Studio 1.5.1 Build 141.2456560 [25] was used to develop the project. The minimum SDK version required for this application development is 21; the minimum complier version required to build this application is 23 and the build tool version is 22.0.1. Apart from the GUI-related code, there are four prominent code portions categorized as 1) Twitter Integration with EasyAuth, 2) Voice authentication using Vigo and its operations, 3) Fingerprint

Authentication using SPass API and, 4) OTP Generation using web-service. The section below describes about the important code snippets of the EasyAuth application.

A. Twitter Integration with EasyAuth

1) Code for integration with Twiiter.com – Twitter Login

```
private void loginToTwitter() {
    boolean isLoggedIn =
mSharedPreferences.getBoolean(PREF_KEY_TWITTER_LOGIN,
false);
    if (!isLoggedIn) {
        final ConfigurationBuilder builder = new
ConfigurationBuilder();
        builder.setOAuthConsumerKey(consumerKey);
        builder.setOAuthConsumerSecret(consumerSecret);
        final twitter4j.conf.Configuration configuration =
builder.build();
        final TwitterFactory factory = new
TwitterFactory(configuration);
        twitter = factory.getInstance();
        try {
            requestToken =
twitter.getOAuthRequestToken(callbackUrl);
            final Intent intent = new Intent(this,
WebViewActivity.class);
            intent.putExtra(WebViewActivity.EXTRA_URL,
requestToken.getAuthenticationURL());
            startActivityForResult(intent,
WEBVIEW_REQUEST_CODE);
        } catch (TwitterException e) {
            e.printStackTrace();
        }
    } else {
        loginLayout.setVisibility(View.GONE);
        shareLayout.setVisibility(View.VISIBLE);
    }
}
```

This function contains functionality regarding twitter's login. It is using Twitter4j library to perform the login. The first line checks the applications' local shared preferences to check if a user of twitter is already logged in. We have stored the flag regarding twitter login as true or false to app's private context. So, if we get that false, it will attempt for the twitter login with a consumer key and secret provided by twitter developer console. In case of requesting twitter's API, developers need to request twitter API with consumer key and consumer secret. To get consumer key and consumer secret, developers need to create an app here [26] and fill the details regarding the app. Once it is successfully created it will obtain Consumer Key and Consumer Secret which will be used to send request to twitter's API. The above method creates the instance of twitter library and sends request for authentication token to twitter's API using the provided consumer key and secret. Once the Auth token is obtained, the user login will be successful.

2) Initialize Twitter Configuration

```
private void initTwitterConfigs() {
    consumerKey =
getString(R.string.twitter_consumer_key);
    consumerSecret =
getString(R.string.twitter_consumer_secret);
    callbackUrl = getString(R.string.twitter_callback);
    oAuthVerifier =
getString(R.string.twitter_oauth_verifier);
}
```

To request twitter's API, we need for authentication, we need to use a Consumer Key and Consumer Secret, which can be obtained by creating application at twitter's developer console [27]. We have created new app at twitter's developer console, completed the app creation process and obtained the Consumer Key and Consumer Secret to integrate in our Android app. Also, to use these configuration details throughout the application, we have stored the values to the string.xml file. From there we can get values by calling a getString(int ID) function. In this way, we have initialized the local String variables before calling loginTwitter( ) function.

3) Save Twitter Configuration to local shared preferences

```
private void saveTwitterInfo(AccessToken accessToken) {
     long userID = accessToken.getUserId();
     User user;
     try {
         user = twitter.showUser(userID);
         String username = user.getName();
         SharedPreferences.Editor e =
mSharedPreferences.edit();
         e.putString(PREF_KEY_OAUTH_TOKEN,
accessToken.getToken());
         e.putString(PREF_KEY_OAUTH_SECRET,
accessToken.getTokenSecret());
         e.putBoolean(PREF_KEY_TWITTER_LOGIN, true);
         e.putString(PREF_USER_NAME, username);
         e.commit();
     } catch (TwitterException e1) {
         e1.printStackTrace();
     }
 }
```

This function is used to save the twitter auth token and auth secret as well as other profile information to the application's private shared preferences. Once the twitter authentication is completed successfully, the API will return the **oAuthToken** and **oAuthSecret,** which will be used to fetch other details regarding logged-in users. Once we get all details, we call a saveTwitterInfo function to save all details to private context. Those can be used globally throughout the application in case of fetching details regarding logged-in user's details. We also need to display logged in user's tweets once the authentication is completed successfully. So, in that case we can use oAuthToken and oAuthSecret to request twitter's API and fetch user's tweets and display to the screen. In case of this, we use these values globally in the app; we have stored the information to application's shared preference using this function.

B. Voice Authentication using Vigo and its operations

1) Code for initialization using Vigo credentials

```
ViGoLibrary.getInstance().init(
        VIGO_CREDENTIAL_ID,
        VIGO_CREDENTIAL_PWD,
        VIGO_SERVER_URL,
        VIGO_APP_ID);
```

In case of voice authentication, we need to use Vigo Library. The above code is used for initialization of the library. To request Vigo API, we need CREDENTIAL_ID, CREDENTIAL_PASSWORD, URL and APP_ID. To get these details we need to complete registration at Voice Vault [28]. Once we get all the details, we need to integrate them to

application and using these we can send request to Vigo Library. It is important to note that the Vigo Library is paid but provides a 45 days' trial version, which we have used for our development.

2) Code for Register User using Vigo library

```
public void startRegistrationClick(View button) {
    button.setEnabled(false);
    if (mClaimantId == null) {
        ViGoLibrary.getInstance().registerClaimant(this)
    }
    else if (!mIsClaimantRegistered) {
        registerClaimantCallback(mClaimantId);
    }
}
```

In case of using Voice Authentication, we need to register the voice with Vigo Library before using login. To register the voice with Vigo Library the above code snippet will be used. Also, when registration is completed successfully, the registered call back event will get notified and the details will be passed to a new recording screen. Once the user is registered with Vigo Library, the claimant ID will be provided to users and that will be used to record an audio phrase with Vigo Library.

3) Code to record voice using Vigo library

```
public void recordClick(View recordButton) {
    findViewById(R.id.buttonRecord).setEnabled(false);

findViewById(R.id.buttonRecord).setSoundEffectsEnabled(false)
;
mTextViewStatus.setText(getString(R.string.status_recording))
;
    ViGoLibrary.getInstance().startRecording(
            VIGO_RECORD_TIME_MILLISECS,
isAudioRecordVoiceRecognitionOptionEnabled, this);
}
```

In the case of voice authentication, the audio should be registered to Vigo API. The above displayed function is used to record a user's voice with Vigo Library. The method is useful to both REGISTER YOUR VOICE and LOGIN WITH YOUR VOICE cases. There is a method with Vigo Library to record user's voice, startRecording () which will require 3 arguments: Time in millisecond (VIGO_RECORD_TIME_MILLISECS), in or case 4000 milliseconds (4 seconds) in our case; a flag to check whether the voice recognition feature is available to the device or not; and context of activity class. Vigo Library has an interface VoiceVaultAPIVoiceCallback, which contains the callback methods like onRecordCompleted (), which will be notified when a recording is completed after 4 seconds. Thus, this way voice authentication is established as an authentication scheme in EasyAuth application.

C. Fingerprint Authentication using SPass API

```
private SpassFingerprint.IdentifyListener mIdentifyListener
= new SpassFingerprint.IdentifyListener() {
    @Override
    public void onFinished(int eventStatus) {
        int FingerprintIndex = 0;
        String FingerprintGuideText = null;
        try {
            FingerprintIndex =
mSpassFingerprint.getIdentifiedFingerprintIndex();
        } catch (IllegalStateException ise) {
        }
        if (eventStatus ==
SpassFingerprint.STATUS_AUTHENTIFICATION_SUCCESS) {
            Toast.makeText(DashBoardFingerPrint.this,
"Success",Toast.LENGTH_SHORT).show();
            Intent intent = new
Intent(DashBoardFingerPrint.this , DashBoard.class);
            startActivity(intent);
            finish();
        } else if (eventStatus ==
SpassFingerprint.STATUS_AUTHENTIFICATION_PASSWORD_SUCCESS) {
        } else if (eventStatus ==
SpassFingerprint.STATUS_OPERATION_DENIED) {
        } else if (eventStatus ==
SpassFingerprint.STATUS_USER_CANCELLED) {
        } else if (eventStatus ==
SpassFingerprint.STATUS_TIMEOUT_FAILED) {
        } else if (eventStatus ==
SpassFingerprint.STATUS_QUALITY_FAILED) {
            needRetryIdentify = true;
            FingerprintGuideText =
mSpassFingerprint.getGuideForPoorQuality();
            Toast.makeText(DashBoardFingerPrint.this,
FingerprintGuideText, Toast.LENGTH_SHORT).show();
        } else {
            needRetryIdentify = true;
        }
        if (!needRetryIdentify) {
            resetIdentifyIndex();
        }
    }
    @Override
    public void onReady() {
    }
    @Override
    public void onStarted() {
    }
    @Override
    public void onCompleted() {
        onReadyIdentify = false;
        if (needRetryIdentify) {
            needRetryIdentify = false;
            mHandler.sendEmptyMessageDelayed(MSG_AUTH, 100);
        }
    }
};
```

In case of the fingerprint authentication, users need to register the fingerprints with SPass API. After registration is successful, it will start identifying the fingerprint and compare with a registered one. The above code snippet is showing the listener for the identification process. There are various methods that are called in various cases which are explained below. 1) onReady - This will notify when the SPass library is ready for the identification process. 2) onStart - This will notify when the user starts the identification process.3) onFinished - This will notify when the user finishes the identification process. 4) onCompleted - This will get notified once the whole process finished. Here we are attempting to retry if identification is not successful. The requirement for fingerprint authentication is a smart phone with an inbuilt sensor to make use of this authentication scheme.

D. OTP Authentication using web service

```
private void startTimer() {
    countDownTimer = new
```

```
CountDownTimer(totalTimeCountInMilliseconds, 500) {
        @Override
        public void onTick(long leftTimeInMilliseconds) {
            long seconds = leftTimeInMilliseconds / 1000;
            if (leftTimeInMilliseconds <
timeBlinkInMilliseconds) {

timer.setTextAppearance(getApplicationContext(),
R.style.blinkText);
                if (blink) {
                    timer.setVisibility(View.VISIBLE);
                } else {
                    timer.setVisibility(View.INVISIBLE);
                }
                blink = !blink;
            }
            remainingSecond = leftTimeInMilliseconds;
            Log.i("", "++" + remainingSecond);
            timer.setText(String.format("%02d", seconds /
60)
                    + ":" + String.format("%02d", seconds
% 60));
        }
        @Override
        public void onFinish() {
            Intent intent = new
Intent(CodeVerificationActivity.this,
ResendCodeVerification.class);
            intent.putExtra("emailId", emailId);
            startActivity(intent);
            finish();
        }
    }.start();
}
```

In case of using OTP Authentication, we need to generate One Time Password to complete the process. Once the OTP is generated, it will be sent to our server and will also be sent to users via email; valid for 2 minutes. The above method is used to display the timer for the OTP lifetime on screen. Once the timer is over, the user will be redirected to another screen with the message "The OTP which was sent to you in email is expired". If desired, users can resend the OTP again by clicking on Resend OTP.

```
Random rnd = new Random();
n = String.valueOf(100000 + rnd.nextInt(900000));
```

The above code snippet is used to generate a new random number between 100000 and 900000. We can change the range also and generate more complex codes that contain alpha numeric characters and special symbols randomizing it as required to make the guesswork difficult for hackers.

IV. Test Results For EasyAuth

We took several test cases and tested the EasyAuth Application in various different environments and the entire performance analysis is shown on graphs. They are plotted by taking fixed number of counts and later on we are able to derive which authentication scheme is accurate and which one fails when the environment changes.

Fig. 2 Test Results in various environments for Voice and Fingerprint

Figure 2 depicts pass as well as fail scenarios for all the different test environments we choose for testing this application. We tested this application in environments which include: authentication with voice based authentications schemes at a coffee shop, while driving a car, while driving a car with loud music, while driving a car with slow music, while walking, while cooking and other noise heavy environments. We have observed that the voice authentication takes a longer time during loud music and also if one is using a mobile network like 4G/LTE, it will sometimes fail due to the lack of network availability. For fingerprint authentication, we took test cases like touching the sensor with a wet finger, applying little oil on the finger, applying talcum powder on finger and so on. We have observed that it fails to recognize your fingerprint if we consider such scenarios.

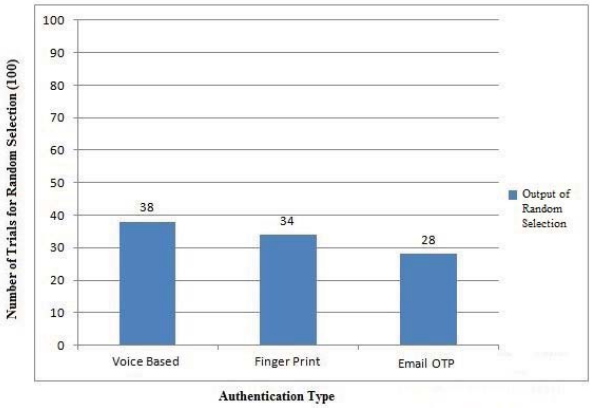


Fig. *3 Test Results which shows Random Selection given by System for three authentication schemes*

Figure 3 depicts the relationship between the number of trials (Total 100), which we selected for testing versus different authentication schemes. Whenever a user selects all three authentication schemes, the system would give any one out of three whenever a user tries to login to Twitter. We tried to login 100 times, out of which 34 times the system gave us fingerprint authentication, 28 times OTP and 38 times voice. Thus, the voice authentication scheme topped among the three during our testing.

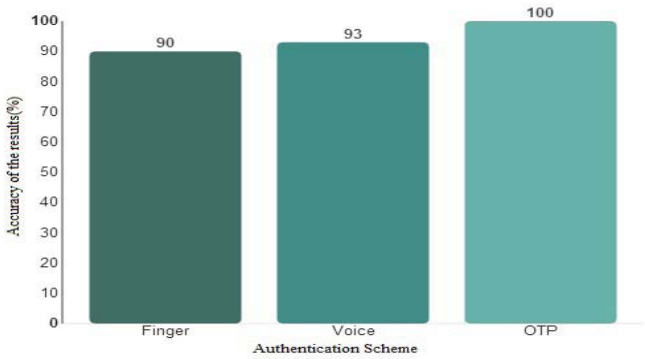


Fig. *4 Comparison of Accuracy of Voice, Fingerprint and OTP Scheme*

Implementing three authentication schemes along with a single factor authentication gave us multi-factor authentication systems especially for Twitter.com but it was quite challenging to figure out which one is more accurate. For this testing, we chose fixed counts of trails to login to Twitter but took several test cases to measure the accuracy of EasyAuth.

From the Figure 4, one can make out that of all three mechanisms, OTP is always successful if user enters correct code unless a wrong code or wrong email address is provided. To conclude, for voice it failed during loud music environments specifically and thus is 93% accurate and for fingerprint we intentionally tested by applying talcum powder as well as water on finger which resulted in low accuracy specifically for our testing around 90%.

V. Conclusion

We have shown that by implementing multi-factor authentication, we can surely make the login process of Twitter.com easier and also provide users a hassle-free login to Twitter by using "something they are" as well as "something they know", like voice, fingerprint and OTP. We also provide users the flexibility to choose the level of desired security as per their preference so users can be flexible. Our results show that multi-factor authentication is much better; and by randomizing it with different authentication schemes, can be used effectively which makes the system sustainable and secured against the guesswork of hackers. Many issues were discovered with the Android fingerprint scanner which comes inbuilt so we have aimed to resolve the problem of unrecognized fingerprints. We also target to develop an iOS version of the EasyAuth application in future.

VI. References

[1] Password Rage. http://www.information-age.com/technology/security/123459599/do-you-have-password-rage-third-people-admit-tantrums-over-password-frustration
[2] Vigo Architecture and Principles. http://voicevault.com/wp-content/uploads/2014/03/ViGo-Architecture-and-Principles.pdf
[3] Vigo Introduction & Security. http://voicevault.com/wp-content/uploads/2014/03/ViGo-Introduction_secured.pdf
[4] What is so unique about your fingerprint? http://wonderopolis.org/wonder/what-s-so-special-about-your-fingerprints
[5] One-Time Passwords – htop and totp. http://blogs.forgerock.org/petermajor/2014/02/one-time-passwords-hotp-and-totp/
[6] Google Authenticator Support Answers. https://support.google.com/accounts/answer/185833?hl=en
[7] EMC INC. RSA Secure ID. http://www.emc.com/collateral/data-sheet/h13821-ds-rsa-securid-hardware-tokens.pdf
[8] Yubi Key Hardware. https://www.yubico.com/faq/yubikey/
[9] Fido U2F Specifications. https://fidoalliance.org/specifications/overview/
[10] Google 2-Step Verification. http://www.google.com/landing/2step/
[11] Google Authenticator – Product Forum Topic. https://productforums.google.com/forum/#!topic/gmail/4-D-0lXGtwc
[12] 2-factor authentication v/s Multi-Factor Authentication. http://mi-token.com/2fa-vs-multi-factor-authentication/
[13] Multi-factor Authentication Explanation with Authentication Factors. http://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA
[14] Impermium study unearths consumer attitudes toward internet security. http://goo.gl/NsUCL7, 2013Lagrange polynomial interpolation. http://www2.lawrence.edu/fast/GREGGJ/Math420/Section_3_1.pdf
[15] PETSAS, T., TSIRANTONAKIS, G., ATHANASOPOULOS, E., AND IOANNIDIS, S. Two-factor authentication: Is the world ready? Quantifying 2FA adoption. In 8th European Workshop on System Security (2015), EuroSec '15
[16] GUNSON, N., MARSHALL, D., MORTON, H., AND JACK, M. A. User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. Computers & Security 30, 4 (2011), 208–220.
[17] WEIR, C. S., DOUGLAS, G., RICHARDSON, T., AND JACK,M. A. Usable security: User preferences for authentication methods in e-banking and the effects of experience. Interacting with Computers 22, 3 (2010), 153–164.
[18] Multi-factor authentication using Microsoft Azure. https://azure.microsoft.com/en-us/documentation/articles/multi-factor-authentication/
[19] Twitter 4j API. http://twitter4j.org/en/powered-by.html
[20] Twitter Accounts Hacked in 2013- Report in Abcnews. http://abcnews.go.com/blogs/technology/2013/02/250000-twitter-accounts-hacked-dont-panic-heres-what-to-do/
[21] Twitter Accouts Vulnerable in 2013. https://www.thewrap.com/twitter-warns-users-about-hacked-accounts/
[22] Bug in Twitter. https://blog.twitter.com/2016/fixing-a-recent-password-recovery-issue
[23] Vigo Rest API Guide. http://voicevault.com/wp-content/uploads/2014/03/ViGo-REST-API-Guide.pdf
[24] Samsung Pass API Documentation. http://img-developer.samsung.com/onlinedocs/sms/pass/index.html
[25] Android Studio. http://developer.android.com/tools/studio/index.html
[26] Developer Twitter for new application. dev.twitter.com/apps/new
[27] Create Application at Twitter Console. https://apps.twitter.com/
[28] Vigo Free Trail Registration Sign up Link. http://voicevault.com/for-developers/#signup
[29] Comparison of Top MFA products. http://searchsecurity.techtarget.com/feature/The-fundamentals-of-MFA-Comparing-the-top-multifactor-authentication-products
[30] Last Pass. https://lastpass.com/support.php?cmd=showfaq&id=375
[31] MFA Heads PCI's List of Change. http://www.paymentssource.com/news/retail-acquiring/multi-factor-authentication-heads-pcis-list-of-changes-3023992-1.html

Implementation of the Enhanced Fingerprint Authentication in the ATM System Using ATmega128 with GSM Feedback Mechanism

Kennedy Okokpujie[1], Funminiyi Olajide[2], Samuel John[3], Chinyere Grace Kennedy[4]
[1,3]Department of Electrical and Information Engineering
[2]Department of Computer and Information Sciences
Covenant University, Ota, Ogun State. Nigeria.
[4]Dept. of Computer Science and Engineering,
Ewha Womans University, Seoul, South Korea
[1]kennedy.okokpujie@covenantuniversity.edu.ng; [2]funminiyi.olajide@cu.edu.ng; [3]samuel.john@covenantuniversity.edu.ng,
[4]gkennedy@ewhain.net

Abstract- *ATM was introduced to boost the cashless policy in Nigeria. Current trend of Cybercrime facilitate the need for an enhanced fingerprint application on ATM machine with GSM Feedback mechanism. The mechanism enable unassigned fingerprint authentication of customers with quick code and secret code. The project enhances the security authentication of customers using ATM. A core controller using fingerprint recognition system of ATmega128 in-system programmable flash is explored. An SM630 fingerprint module is used to capture fingerprints with DSP processor and optical sensor for verification, using AT command of GSM module for feedback text messaging (i.e. sending of Quick and Secret-Codes respectively). Upon system testing of capable reduction of ATM fraud using C program, the new method of authentication is presented.*

Keyword- *Automated Teller Machine (ATM) , ATmega128, GSM Module, Language C program, SM360 Fingerprint Module*

1.0 Introduction

The ATM card and PIN have proven to be inadequate security due to the continuous rising threat of ATM related frauds in the emerging global cashless economy. For instance in Nigeria, there are various security breaches and ATM related fraud has risen from 1.6 billion naira (10 million USD) in 2010 to 40 billion naira (250 million USD) in 2013. However, some other countries have higher figures [12]. With technology advancement in electronic banking, bank customers have embraced the use of ATM (Automated Teller Machine) being a unique banking product in Nigeria.

Data from the Nigeria Inter-Bank Settlement System (NIBSS) has revealed that the highest number of fraudulent transactions in the banking sector takes place on Automated Teller Machines (ATMs) with 43 per cent of electronic banking frauds, followed by internet banking which was responsible for 34 per cent [9]. The NIBSS data showed that three per cent of electronic banking fraud took place on Point of Sales (PoS) terminal while e-commerce was responsible for one per cent of electronic banking fraud and others, 19 per cent.

ATM is an electronic gadget that has its roots embedded in the accounts and records of a banking institution [5]. It is a machine that allows the bank's customers to carry out banking transactions like; cash transfers between or among same bank (intra bank-transfer), between or among different bank (inter-bank transfer), between or among a bank customer account and a mobile phone account (mobile money transfer), account balance enquiries, payment of utility bills (electricity bill, water rate, etc.), recharging of air time, cable bill payment, governments levy (Vehicle particulars, custom duties, tenement rate, import and export duty, personal tax income) and cash withdrawal. ATM machine with its 24hours availability also allows those who have no access to internet to carry out their transactions anytime but on real time online platform. This operation has led to ever increasing demand of ATM services been rendered by banks.

Traditional authentication systems (use of PIN) cannot discriminate between an impostor who fraudulently obtains the access privileges (card and PIN) and the genuine user [1][6]. Therefore, to gain the ATM machine user's confidence, a second level biometric authentication security has to be put in place in conjunction with the already existing personal identification number (PIN). The activities of ATM fraudsters in Nigeria have brought about financial hardship and devastation to victims and their families. These activities can also have negative effects on a nation's economy and has cause the erosion of trust of banking institutions by the banking public. Hence, there is need to urgently tackle this problem.

2.0 Design Consideration and Specification

The embedded ATM client verification system is based on fingerprint recognition which is designed to improve on the performance of the existing ATM system. The ATmega128 chip is used as the core of this embedded system which is associated with the technologies of fingerprint recognition, GSM feedback mechanism and current high speed network communication. The primary features of the developed system are:

- Fingerprint recognition: The masters' fingerprint information was used as the standards of detection. It must certify the feature of the human fingerprint before using ATM system.
- Remote verification: System that can compare current client's fingerprint information with remote fingerprint data server.
- GSM: There is customer's secret code (i.e. S-Code) generated upon registration of fingerprint in the bank. In an exception of fingerprint verification error of a genuine user, the system demands the customer secret code. In order not to deny a genuine customers access into his/her account, the system is capable to quickly generate a unique 4-digit access code (i.e. Q-Code) on a condition that the customer supplied a correct secret code. This 4-digit access code will be sent as OTP (One Time Password) message code to mobile phone of the authorized customer.
- Two discriminate analysis systems: Unimodal Biometric and Two-tier Security. Two-tier security is used to provide two levels of security. In unimodal system, if the fingerprint system fails (this situation happens very rarely) then, two level security units will take over and further queries will be required from such a user.

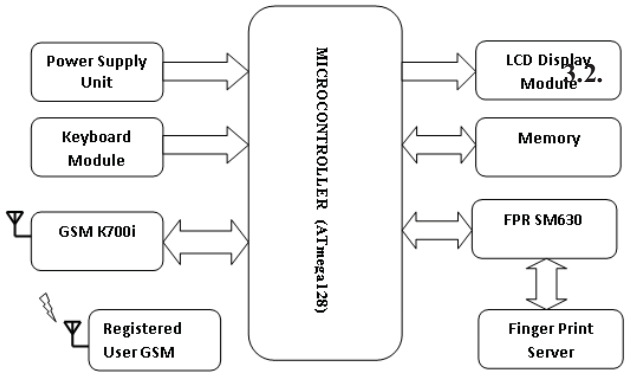


Figure 1.0: Block diagram of the designed system

3.0 Design Analysis of different Sections of the System

The design and implementation of the security for ATM terminals system consist of two parts which are hardware and software. The hardware is designed by the rule of embedded system and the aspect of software consists of several parts [7]. Figure 1 shows the major system modules and their interconnections.

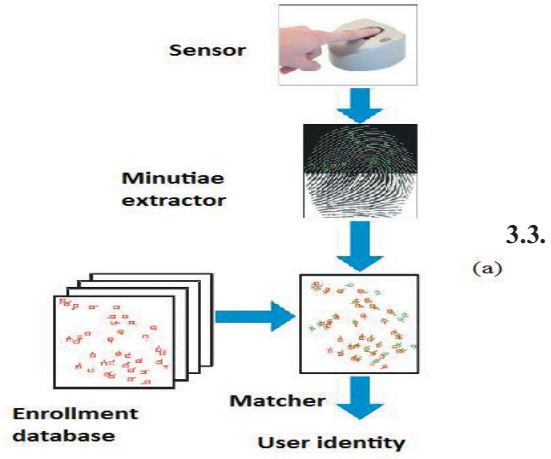
3.1. Microcontroller (ATMEGA128)

The system uses ATmega128 from ATMEL family; it is the core controller in the system. ATmega128 is an 8-bit Atmel microcontroller with 128Kbytes in-system programmable flash with advanced RISC (Reduced Instruction Set Computer) Architecture of 32 x 8 general purpose working register plus peripheral control registers with full static operation. It offers high performance for very low power consumption and cost. The Atmel architecture is based on RISC principles, and the instruction set and related decoding mechanism are much simpler than those of micro-programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput up to 16MIPS at 16MHz and an impressive real-time interrupt response from a small and cost-effective chip. (Amtel Microcontroller, 2011)

The Atmel memory interface has been designed to allow the performance potential to be realized without incurring high costs in the memory system. Speed-critical control signals are pipelined to allow system control functions to be implemented in standard low-power logic, and these control signals facilitate the exploitation of the fast local access modes offered by industry standard dynamic RAMs. The ATmega128 device is supported with a full suite of program and system developed tools including: C compilers, macro assemblers, program debuggers/simulators, in-circuit emulators and evaluation kits. These made it suitable for the actualization of the project.

3.2 Fingerprint Module (SM630)

The communication with the fingerprint module is made through (RXD0/PDI) PE0 port [2] and (TXD0/PDO) PE1 port [3] via UART0 of ATmega128. A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. [14]. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching as shown in Figure 2[2].



3.3. (a)

Figure 2: A typical automated fingerprint recognition system,

Evolutionary standalone fingerprint recognition module SM630 is ideal for on-line applications because it allows ASCII commands to manage the device from the host. On-line functionality, can verify fingerprints and them store on non-volatile memory. The most important module of the system is the fingerprint scanner. The SM630 by Miaxis was used. It consists of optical fingerprint sensor, high performance DSP processor and flash. It boasts of functions such as fingerprint login, fingerprint deletion, fingerprint verification, fingerprint upload, fingerprint download etc. [8]

The SM630 has the following unique features hence was used to actualize this project:

- High Adaptation to Fingerprints: When reading fingerprint images, it has self-adaptive parameter adjustment mechanism, which improves imaging quality for both dry and wet fingers. It can be applied to a wider public.
- Algorithm with Excellent Performance: SM630 module algorithm is specially designed according to the image generation theory of the optical fingerprint collection device. It has excellent correction & tolerance to deformed and poor-quality fingerprint.
- Easy to Use and Expand and Low Power Consumption: operational current <80mA.
- Integrated Design: Fingerprint processing components and fingerprint collection components are integrated in the same module. The size is small and there are only 4 cables connecting with HOST, much easier for installation. The operating Voltage: 4.3V～6V, Operating Current : <80mA（Input voltage 5V）,Power-on Time : <200ms（Time lapse between module power-on to module ready to receive instructions, Tolerated Angle Offset : ±45°, User Flash Memory : 64Kbyte, Interface Protocol: Standard serial interface （TTL level, Communication Baud Rate: 57600bps Operating Environment: Temperature: -10℃～+40℃ and Relative humidity: 40％RH～85％RH（no dew).

3.3 GSM MODULE

Global System for Mobile Communications (GSM) is the most popular standard for mobile phones in the world. GSM has over two billion users worldwide and is available in over 213 countries and GSM represents 82.4% of all global mobile connections. GSM uses a variation of Time Division Multiple Access(TDMA) and GSM is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band[4].

The GSM module used in implementing this system is Sony Ericsson K700i; IC4170B-A1021041. K700i GSM engines operate in the GSM 900 MHz, 1800MHz and 1900MHz frequency bands. As shown in Figure 3 , we do not replace the PIN verification. If PIN is correctly entered, system will capture it and match fingerprint of the customer . But if fingerprint does not match, due to any challenge with the fingerprint reader or the customer finger, the system will give the customer opportunity to enter his sceret code. If this is correctly entered, a Quick code (Q-code) is generated by the system and it is sent to the customer mobile phone.When the Q-code of the user is correctly entered the customer will be able to carry out banking transactions otherwise the customer account is block upon three unsucessful attempts. The security provided by this system is foolproof.

3.4 User Interface

The user interface includes the input and output devices and this makes communication with the system easier. An Alphanumeric LCD Display was used with the following features: 4-bit mode – 4-bit (nibble) data transfer and does not use DB0-DB3 – Each byte transfer is done in two steps: high order nibble, then low order nibble, interface requires only 7 I/O pins of microcontroller (DD4-DB7, RS, R/W and E). It is a 20 by 4 character-only LCD display with four character line and 20 characters per line.The matrix keypad consists of several buttons which are arranged in a matrix array for 4x4 keys and interfaced with ATmega120 through GPIO of PC0 (A8) port through PC7 (A15) port. [3]

3.5 Power Supply

This section supplies power to all the sections mentioned above. It basically consists of a transformer to step down the 220V ac to 15V ac followed by a diodes bridge-rectifier. After rectification process, the obtained rippled dc is filtered using a capacitor filter of C=1000μF. A positive voltage of 3.5V and 5V are made available through LM317T and LM7805 to various on board components.

4.0 Software Design

The embedded platform discussed aboved is programmed in language C with AVR studio to follow the program logic as shown in Figure 3.

4.1 Using AVR Studio For C Programming.

AVR Studio is a large piece of software, it supports several of the phases required when programming the ATmega128 microcontroller. AVR Studio is an Integrated Development Environment (IDE) for writing and debugging AVR applications in Windows 9x/ME/NT/2000/XP/VISTA /WIN 7 environments. AVR Studio provideded us with a project management tool, source file editor, simulator,assembler and front-end for C/C++, programming, emulation and on-chip debugging. AVR Studio supports the design, development, debugging and verification aspects of the system.
In programming the ATmega128 microcontroller, four major stages were involved:

- *Create an Atmel Studio project,*
- *Compile C code to produce a HEX file,*
- *Debugge C program using the simulator,*
- *Download HEX file to the STK500 development board and running it.[12]*

4.2Embedded Language C

The AVR Studio 4 platform put forward the options for assembly language and high level language programming. C language being the most convenient language to access different port pins of ATmega128, we programmed the algorithm to control the SM630 fingerprint module through host controller ATmega128 in C language. The program follows the control actions as shown in Figure 3. The program segments to access UART, LCD, RTC, ADC, DAC, are included by linking through UART0.h, LCD.h, RTC.h, ADC.h, DAC.h header files respectively.

5.0 Design Process of the Fingerprint ATM System.

The construction started with the circuit design and this was accomplished with the labcenter Proteus software. Circuit was drawn and double checked, however in order to simulate the performance of the circuit, the software for the atmega128 controller has to be developed first. The design environment was therefore switched over to Atmel AVR-Studio. The AVR-Studio is a software development and debugging environment for the Atmel AVR microcontroller family, to which the atmega128 belongs.

The Atmel AVR-Studio however relies on AVR-GCC compiler for its code compilation. Codes were then developed and compiled in the studio. After successful compilation, codes were then imported into Proteus for simulation. Debugging the code henceforth, involves switching back and forth between AVR-Studio to edit the code and Proteus to simulate it.

On completion of the simulation, the PCB design was done using the Proteus Ares package. The Ares autoplacer and autorouter was used to design the PCB. After routing the PCB, the fabrication was then done using the toner heat transfer method and followed by etching in a solution of hydrochloric acid. After etching, the board was drilled and components mounted appropriately.

6.0. Operational Principle of the Designed System.

As shown in Figure 3, researcher do not replace the PIN verification. If PIN is correctly entered, system will capture it and match fingerprint of the customer. But if fingerprint does not match, due to any errors with the fingerprint reader or the customer finger, the system will give the customer opportunity to enter his sceret code. If this is correctly entered, a Quick code (Q-code) is generated by the system and it is sent to the customer mobile phone. When the Q-code of the user is correctly entered, customer can carry out banking transactions otherwise, the customer account is block upon three unsucessful attempts. The security provided by this system is foolproof.

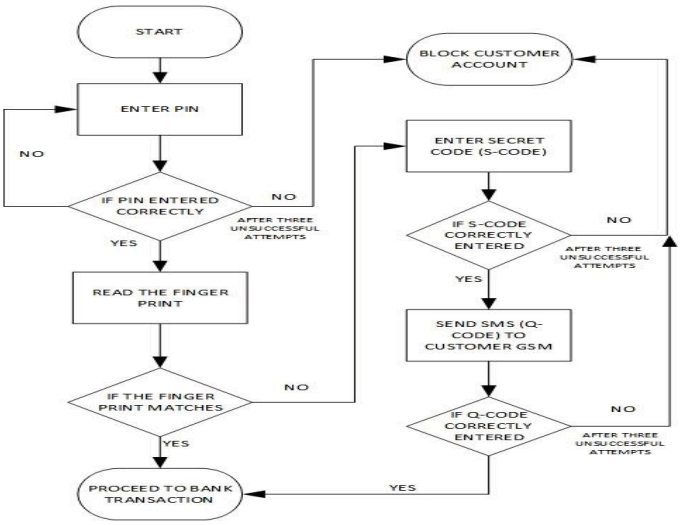


Figure 3: Software designed and flow chart of the designed system.

7.0 Testing Of The Biometric Atm System.

The testing of the designed system was carried out in an academic environment and with the following sample of customers database in Table 1.0 was created by the system. The system is capable of generating and assigning account to the newly registered customer on suppling following customer detail via the system administrator menu in this format. For example:

Name= Mike Joseph
Contact=8253325661
Pin=5201
Balance=7000
Secret=2211

On receiving this customer information by the system, the system is programmed to automatically generate an account number for the customer information and then sent forth a text message to the customer's registered mobile phone (e.g 8253325661) in this format type:

KKK bank alert
Account name: Mike Joseph
Account number:1234620062
Balance:N7,000.00K

On receiving this information, the customer goes to the bank and input the account information on the biometric ATM, and also register his right thumb with the account. This is verified by the bank officer. This completes the registration process of the new customer using the designed Enhenced Biomteric ATM with GSM feedback Mechanism. The information from one thousand (1000) different customers were processed and subjected to testing as sample shown in Table 1.0.

Table 1.0: Sample of Customer database generated by the designed system

S/N	ACCOUNT NAMES	ACCOUNT NO.	REGD. GSM NO.	DATE CREATED	PIN	OPENING BALANCE	S-CODE
1	Mike Joseph	1234200620	8253325661	4/1/2016	5201	N7,000.00K	2211
2	Osato Osaro	1234620062	8123446844	7/1/2016	7342	N4,000.00K	2012
3	Babatude Ola	1234000620	8765432309	7/1/2016	8974	N8,000.00K	1987
4	Obinna Stone	1234000240	8675849302	18/1/2016	1537	N2,000.00K	1642
5	Fred Oba	1234620006	8908765432	21/1/2016	9999	N9,000.00K	1773
6	Abbas clement	1234620064	8432567189	3/2/2016	2387	N3,000.00k	3879
…	…	…	…	…	…	…	…
1000	Oluwa Light	1234000622	8876994321	9/1/2015	2496	N5,000.00K	4382

8.0 RESULT PRESENTATION AND ANALYSIS.

Fingerprint identification system performance is measured in terms of the following parameters and were used to analyze the result of the designed biomteric ATM system [10][11].

A. False Rejection Rate (FRR): The probability that a system will fail to identify an enrollee. It is also called type 1 error rate. This is as well known as false nonmatch rate (FNMR).

FRR = NFR÷NEIA = 0 ÷ 1000 = 0

NFR = number of false rejection rates = 0
NEIA = number of enrollee identification attempt = 1000

B. False Acceptance Rate (FAR): The probability that a system will incorrectly identify an individual or will fail to reject an imposter. It is also called as type 2 error rate. This is as well known as false match rate (FMR).

FAR = NFA÷NIIA = 0 ÷ 550 = 0

NFA = number of false acceptance = 0
NIIA = number of imposter identification attempts = 550

C. Response Time (RT): The time period required by a biometric system to return a decision on identification of a sample. The average response time of the designed system is 1.5 seconds.

D. Decision Threshold (DT): The acceptance or rejection of a data is dependent on the match score falling above or below the threshold. The threshold is adjustable so that

the system can be made more or less strict depending on the requirements of any given application. The decision global threshold used in this project is 30

E. Enrollment Time (ET):The time period a person must spend to have his/her fingerprint reference template successfully created. The enrollment time of the designed system is one second.

F. False positive identification rate (FPIR): This occurs when the system finds a hit for a query fingerprint that is not enrolled in the system.

$$\mathbf{FPIR = 1 - (1 - FMR)^N}$$

$$\mathbf{FPIR = 1- (1- 0)^{1000} = 1- (1)^{1000} = 1- 1 = 0}$$

G. False negative identification rate (FNIR): occurs when it finds no hit or a wrong hit for a query fingerprint enrolled in the system. The relationship between these rates is defined by

$$\mathbf{FNIR = 1- (1 - FNMR)^N}$$

$$\mathbf{FNIR = 1- (1-0)^{1000} = 1- (1)^{1000} = 1-1 = 0}$$

where N is the number of users enrolled in the system = 1000.

Where FMR = FNMR = 0 from system testing.

H. Average time of transaction using the designed system, (Normal process time): 50 Seconds.

I. Average time of transaction which using the feedback GSM mechanism.(Q-code and S-code): 2 minutes.

Some other senarios that were experienced in the designed system, for example in the case of using wrong fingerprint thrice for four of the above customers, Q-code were generated and sent to the customers' GSM phone numbers with which they were able to gain access into their accounts only after they have supplied the correct secrete code (S-Code) numbers.

9.0 CONCLUSION

An enhanced biometric ATM with GSM feedback mechanism has been designed, constructed and tested. The proposed system has overcome the limitations that exists in other methods and provides a secured and safe environment that saves the hard earned money of the user. The system has proved to be 95.79% successful from our analysis. The designed system provides an alternative method for verification if the fingerprint operation has a challenge which is via a mobile phone upon correct entering of the customer S-code. The designed system is capable of eliminating ATM fraud.

10.0 REFERENCES

(1) Agbontaen F.O. & Orukpe P. E. "Secured Online Payment using Biometric Identification System". (2013) Advanced Materials Research,Trans Tech Publications, Switzerland Vol. 824 (2013) pp 193-199

(2) Anil K. J., Jianjiang F., Abhishek N. & Karthik N.,(2008) "On Matching Latent Fingerprints," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp.1-8, 2008.

(3) Atmel Microcontroller with 128KBytes In-System Programmable Flash technical maunal © (2011) Atmel corporation, 2467XS–AVR–06/11.

(4) http://www.tech-faq.com/gsm.shtml

(5) Ibidapo, O. Akinyemi, Zaccheous O. Omogbadegun, and Olufemi M. Oyelami (2010)"Towards Designing a Biometric Measure for Enhancing ATM Security in Nigeria EBanking System". International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 06

(6) Khatmode R. P., Kulkarni R.V., Ghodke B. S. Chitte P. P., Anap S. D. (2014) "ARM7 Based Smart ATM Access & Security System Using Fingerprint Recognition & GSM Technology".International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 2, February 2014, pages 856-860.

(7) Mashurano J. & Wang I. (2013) "ATM Systems Authentication Based On Fingerprint Using ARM Cortex-M3". International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 3, March – 2013, pages 1-6.

(8) Miaxis Biometrics Co., Ltd. SM630 Fingerprint Verification Module User Manual 2008-07-01 V1.0.

(9) NIBSS 2015 www.nibss-plc.com.ng accessed 23.12.2015

(10) Pennam K. & Maddhusudhan M.R., (2012) "Implementation of ATM Security by Using Fingerprint recognition and GSM" International Journal of Electronics Communication and Computer Engineering, Volume 3, Issue (1) NCRTCST, ISSN 2249 –071X, pages 83-86.

(11) Pravinthraja S. and Umamaheswari K., (2011) "Multimodal Biometrics for Improving Automatic Teller Machine Security". Bonfring International Journal of Advances in Image Processing, Vol. 1, Special Issue, December 2011, pages 19-25.

(12) Research data on e-fraud in Nigeria by the Financial Institutions Training Centre (FITC), 2014

(13) Ronald J. T., Neal S. W., Gregory L. M., (2009) "Digital Systems: Principles and Applications, 10th Edition", ISBN: 0131725793, published by Pearson Education, Inc.

(14) Vaibhav R. Pandit, Kirti A. Joshi & Narendra G. Bawane (2013) "ATM Terminal Security using Fingerprint Recognition". International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868, pages 14-18.

ISBN: 1-60132-445-6, CSREA Press ©

SESSION

COMPUTER SECURITY + SECURITY APPLICATIONS

Chair(s)

Dr. Martin Zsifkovits
Dr. Devesh Jinwala

ISBN: 1-60132-445-6, CSREA Press ©

Cloud Application Model

Personal Healthcare Record Implementation

Amer Jneid (*Author)*
Arizona State University
Tempe AZ USA
ajneid@asu.edu

Ashraf Gaffar
Arizona State University
Meza AZ USA
ashraf.gaffar@asu.edu

Abstract— Privacy and confidentiality concerns over Personal Health Records (PHR) are on the rise, largely driven by how the Internet is being used as vehicle to transfer such records PHR are generated to the benefit of the user; these records are exchanged through emails, or 3rd party Clouds with little means to protect the privacy of the user or confidentiality of the records.

In this paper we outline some causes of the problem and outline a comprehensive solution that would give users the control to manage their PHR using our cloud application model.

Keywords—Healthcare; Cloud Computing Security; Privacy; Confidentiality; Emerging Technologies and Applications; Database Security; Access Control; Multi-Factor Authentication; Encryption.

I. INTRODUCTION

Most healthcare records can be defined under two categories:

- **PHR** Personal Health Records: These are personal health records that are generated and transmitted to the users to manage and maintain.
- **EHR** or **"EMR"** are **E**lectronic **H**ealth **R**ecords or **E**lectronic **M**edical **R**ecords. These are health records that are generated and maintained by Health Service Providers. In this category the provider has the duty to maintain these records with extreme confidentiality and privacy in Compliance with HIPPA. We mention the most common reasons why PHR are present on various non-secure Internet Clouds (Fig 1):
    1. Remote care: The healthcare industry is gradually adopting remote care and new trends where doctors use Skype and Google hangout calls with patients, causing some PHR records to be exchanged with the caregiver using unsecured internet communications.
    2. Private arbitration: The insurance and healthcare industries have successfully managed to move malpractice dispute resolutions out of the court system into private arbitration. While the proceedings and outcomes of these disputes are private, Personal Health Records are not, because private judges, lawyers and experts are exchanging PHR using unsecured clouds or email systems to perform their job with no consideration for privacy or confidentiality.
    3. Smart phones: Health instruments connected to smart phones or wearable devices are becoming the norm. Measurements such as blood pressure, sugar levels and oxygen levels can be obtained using Bluetooth devices etc. A simple Amazon or Google search reveals thousands of "Internet of Things" ("IoT") Bluetooth and cloud ready devices that generate Personal Health Records, most of which are stored on third party clouds, with no transparency as to their fate. For example, a Bluetooth breathalyzer, while not a medical device, can still generate a PHR that should be treated with privacy and confidentiality.
    4. Some practitioners, hospitals, and private clinics exchange PHR with their patients using non-secure Internet communications.
    5. Judges, lawyers, caregivers and users lack awareness of the risks and privacy implications of transferring PHR via email and open public networks. Even with awareness, currently there are no means available to exchange PHR securely.
    6. The user lacks understanding of the implications of loss of privacy e.g. life insurance policy cost increased, job denial due to confidential health information being leaked, being taken advantage of by intruders, promoters, and social isolation, to name a few. PHR are widely present in various non-secure Clouds. We developed a Cloud Application Model which will protect user privacy and confidentiality against unauthorized access. The model we created consists of layered defense perimeters, fortified by three methods of multi-factor authentication and several layers of encryption. This paper will discuss the model as a solution and then apply it directly to Personal Health Records (Fig 2).

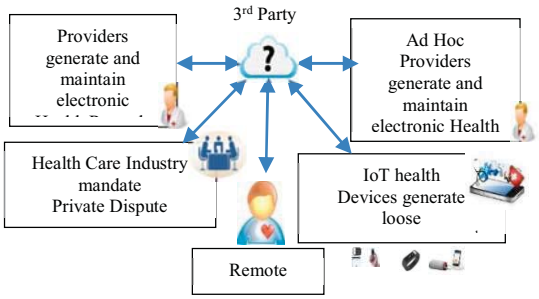


Fig 1 - **CURRENT MODEL – Use of 3rd PARTY CLOUDS PRIVACY AND CONFIDENTIALITY AT RISK**

II. CLOUD APPLICATION MODEL

We built a Cloud Application Model that is applicable to almost any Cloud Application implementation. Fig 2 shows a high-level overview of the solution. In this work we describe how we applied it to PHR. Our Cloud Application Model has four main objectives:

1. Protect privacy and confidentiality
2. Deter hackers by deploying comprehensive security systems that would require costly resources to breach
3. Provide a model for most Cloud Applications
4. Support mobiles and browsers

The Cloud Application Model secures data using 5 layers of defense mechanisms:

a) Protects the Cloud Application with multiple layers of defense perimeters that defend and obscure the data servers and increase hacking cost
b) Fortifies user authentication, preventing hackers from accessing the system using phished legitimate user credentials
c) Encrypts data end-to-end using three methods of encryption to safeguard the data in all three stages: Transfer, In Use, Rested
d) Empowers data owners to manage their information through simple provisioning
e) Creates safe inbound and outbound document sharing to prevent unintended document leaks

Each of those mechanisms is elaborated below.

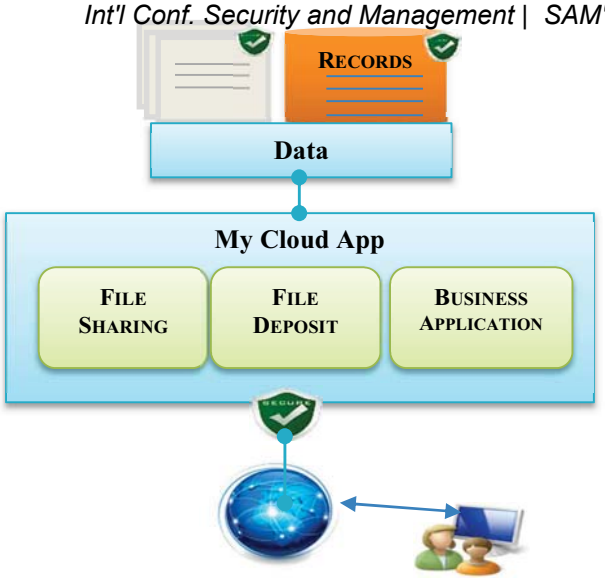


FIG 2 CLOUD APPLICATION MODEL

A. *Cloud App Security*

The Layered Defense Perimeters

1) *Reverse Proxy[1]*

The reverse proxy is the cloud-hosted application point of entry past the firewall. Users interact with the application through this server. From the user perspective, this server appears as if it is the original application server.

Function:

- Handles user's traffic: the server forwards authorized traffic to the server it represents and has no content of its own
- Handles all TLS encryptions for the data being transferred between server and users

Benefits:

- Second line of defense (Firewall being the first)
- Obscures the characteristic of the server next in line
- Offloads encryption of the application server

Vulnerabilities:

Single point of failure: This vulnerability could be mitigated by using an auto-scale, redundant Cloud server.

2) *Intercepting Proxy*

This server forwards traffic received from the reverse proxy to the API server after it enforces users, devices and sessions policies. It is a transparent forward proxy server.

Functions:

- Manages session and related policies
- Manages devices and related policies
- Manages users and related policies
- Extends **L**ight **A**ccess **D**irectory **P**rotocol (LDAP) Authentication to support Multi -Factor Authentication

IBM , Legist LLC *(sponsors)*

- This server knows policies but it does not know anything about the data or the business logic; it forwards compliant traffic to the API server

Benefits:

- Acts as the third line of defense
- Protects the API server from unauthorized requests
- Offloads the API server from policy management
- Obscures the API server

Vulnerability:

Single point of failure: This vulnerability could be mitigated by using an auto-scale, redundant Cloud server.

3) API Server

- Handles data layers
- Handles business logic
- Handles the 2nd and 3rd encryption methods for the data in use and the data at rest. As we described earlier, the 1st encryption is applied to the data in Transfer and it is handled by the reverse Proxy Server. It is important to note that all 3 components use different methods and encryption keys.

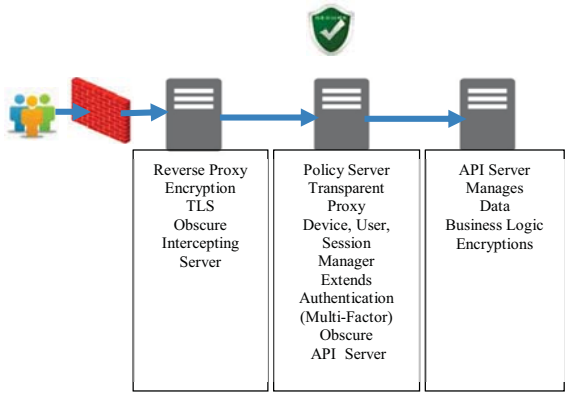


Fig 3: Multi-Layers Perimeters Defenses

B. *Multi- Factor Authentication[12][13][14]*

A significant number of cloud vendors still use One-Factor Authentication methods such as user ID and password to access their cloud service. One-Factor Authentication is based on static knowledge passwords. This makes it popular for two main reasons: a) ease of implementation as it takes very few lines of code to implement, b) ease of use as it takes few seconds and few clicks for authentic users to log in. This false sense of security can be easily breached as both user name and password can be guessed, stolen, found by brute-force or common social engineering methods like phishing. A Cloud with One-Factor Authentication carries significantly higher risk than a Cloud protected with multi-factor authentication. Two Factor Authentication is considered more secure and addresses the weakness of One-Factor Authentication. It is based on using the user ID and password as the first factor and a secret question or one-time password as a second factor. A secret question such as mother's maiden name is a common and probably overused example. A One-Time Password is a password that is valid for only one login session or transaction. For every login the server and the client generate a new, synchronized one-time password based on a shared secret key.

While better than one-factor authentication, two-factor authentication with a secret question is still vulnerable since both factors are static and subject to guessing, phishing or any hacking technique used in one-factor authentication. The only difference would be that it might take a little more time, or experience, to breach.
We recommend the use of Multi-Factor authentication with One Time Password (OTP). There are several types of OTP and several different delivery mechanisms we recommend that use a combination of different types of Authentication methods based on user role.

1. **Method 1-** Hardware Token (Fig 4): requires the user to carry a specialized hardware - not practical for a casual user and for general cloud users. It is good for IT support.

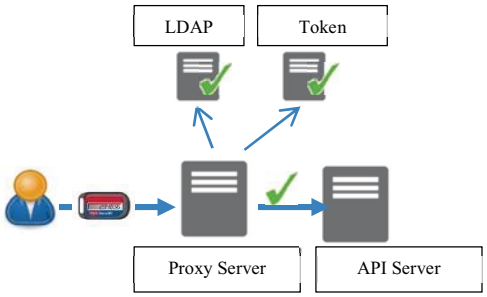


Fig 4 - Hardware Token

2. **Method 2**- Soft OTP sent as an SMS message or email. While they are not ideal these methods cause the least inconvenience. They are also vulnerable to mobile theft. We restrict this method for dynamic or temporary users that use the inbound or direct deposit share.

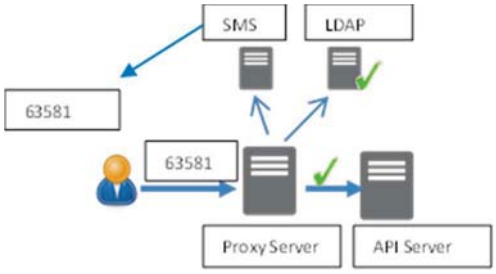


Fig 5 Soft Token

Method 3 [12] - This method is sometimes referred to as device authentication. The device has to be approved. During the approval process a secret key is

negotiated. Each time the user authenticates, it calls the authentication client which will capture from the user his username and password, generates a one-time password that is a function of key and time; sends the server 4 values (user id, user password, device id along with unique OTP). Once the device is approved it acts like standard authentication- no behavioral changes. Unlike methods 1 and 2, the user does not need to enter the OTP. The approved device automatically sends the OTP. This method's limitation is that the user can only access the data from approved devices. One user could have multiple approved devices. It is the most secure approach with fewest behavioral changes.

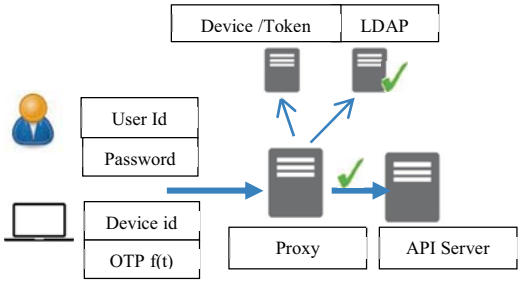


Fig 6 Device and User Authentication

The platform supports all 3 methods- Method 2 we restricted to temporary users. All 3 methods suffer the same vulnerability: stolen device. This could be mitigated by adding geolocation constraint to the mix of policies.

C. Encryption

1) Disk storage encrypted at rest [3][4][5]

Encryption at rest encrypts the data on the physical drive, and only for the authorized users is the data de-encrypted. Encryption at rest is becoming an increasingly accepted method of securing documents. It protects the confidentiality of the data.

Functions:

- Provides client or entity its own set of keys
- Encrypts all files including, documents, logs, database backups, data base files etc

Benefits:

- Protects the confidentiality of documents
- Hard to detect any performance implication
- Increases the hacking cost dramatically

2) Obfuscation for privacy [6][7][8][9][10]

While encryption at rest is essential, it does not protect the data while it is being used. A database engine opens the data files and prepares the data for querying. Therefore, as long as the database engine is running, the data are in a de-encrypted state, which is the case most of the time. To have full protection for databases, the cloud must provide two levels of encryption:

- Encryption at rest to protect the backup files, the logs and the database files
- Encryption of the data before it is stored in the database. However, querying encrypted data is complex. For example, "Hello my name is Peter": encoding this string with 64-bit encryption yields the following string "SGVsbG8gbXkgbmFtZSBpcyBQZXRlcg==" If one wants to search on word "name", one can't simply encrypt the "name" which yields "bmFtZQ==" and perform a search, because this search will fail. With this encryption one has to de-encrypt the entire set and perform the search which defeats the purpose of encrypting the data in the first place; moreover, the operation is slow and impractical for a large dataset. Most encryptions will fail virtually all queries. We developed an algorithm that stores the data obfuscated in the database (Fig 6) with very little resource costs, and near 0 costs in time.

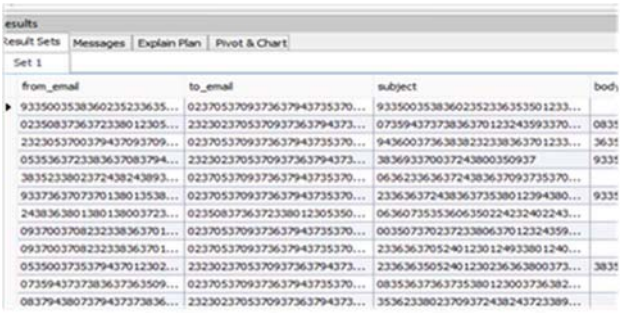


Fig 6 Encrypted Data stored in MySQL database

- All data stored inside the database (users' name, file names, emails) are obfuscated (Fig 6)
- All queries run are unmodified on the obfuscated data including keyword search, caps indifference, except on "sound like" queries that are very hard to do. How does this work? Users provide content to store and the request is securely sent to the API server using TLS. Once it gets to API server, the API server encrypts the data using the proprietary algorithm, and generates the proper insert query but the value is the encrypted data. What happens if the user wants to search on a value? The search request reaches the API server, the API server encrypts the search value and submits a normal select query on the encrypted value. The query result set is encrypted. The API server applies whatever business logic – de-encrypt and had the data back to the proxy server to deliver to the user the information in a readable format. Because the Algorithm is proprietary, we cannot discuss it in this paper.

Vulnerabilities: Brute force on the data set may have a higher success rate. However, this risk is mitigated in the following ways:

1. The encryption at rest protects the files, in case the system was compromised and the files were copied and an offline brute force attack would be quite challenging, as the data is encrypted twice. The decryption is made much more complex.
2. In order for the brute force to work, the hackers have to run queries to siphon the data out of the tables, which could be defeated by monitoring the flow rate of the data, and in our implementation it is very difficult to inject queries because we only allow the API Server to perform database querying. As we demonstrated earlier the API server is well protected. We made it hard for hackers to reach it.

Benefits:

Obfuscation when combined with at rest encryption adds a very good layer of protection for confidentiality against hacking and again increases the hacking cost.

D. Provisioning

The goal of the Model is to make the provisioning process done by the data business owner, not necessarily a skilled IT person. We identified four roles associated with provisioning:

- Data Owner
  - Activate/deactivate users and manage roles
  - Approve devices to access the platform
  - Approve devices and geolocation policies
- User
  - Have access rights to all modules of the business application unless data owner scales some rights back
- Guest
  - Have no rights until data owner grants what she perceives to be appropriate rights and constraints
- Dynamic user
  - Are created dynamically for inbound and outbound documents to share

E. Document Sharing

1) Direct Document Desposit

The point of this implementation is to acquire documents from an outside entity without the use of email or 3rd party clouds' document-sharing.

The requestor must be a valid user verified with the platform. He sends a secure invite to a person to deposit documents. The invite includes a link that once triggered:

a- Sends the person a One Time Token as SMS or email message.

b- Opens a view for the entity to upload one or multiple files at once after entering the valid token.

The person is dynamically created on the system. Once the upload is complete the login for that person is removed off the system. Each invite works only once. The requestor does not have to manage security settings for the person; all provisioning is automated. The requestor received the desired files without the use of email attachments or public shared folders, and is unaffected by the person's environment.

Fig 7 Direct File Deposit

2) Outbound Share

Our file-sharing solution resembles in a lot of ways a standard file-sharing platform with a few exceptions:

1. It utilizes a server-based viewer with the ability to turn on/off printing, cut and paste, and download.
2. For each share we define the following attributes: **D**uration of the share, Cut & Paste=Y/N, **D**ownload Y/N, **P**rint Y/N) a share ID x for person y with D=20, C=N, D=N,P=N means person y has rights to view the shared documents for 20 days only but can't download, print, or cut and paste (Table 1).
3.

Share SI	User Name	View Duration	Download	Print	List of Documents
1	Attorney A	60	N	Y	PDF list
2	Attorney A	45	N	N	PDF list
3	Overseas Provider	5	N	N	PDF list

Table 1- Sample Share Attribute

4. Each time the person clicks on the share link, a temporary login ID and token will be sent to him as an SMS or email message. Once he enters the proper token he can access the share documents assigned to him. The share has a lifetime and once it expires the temporary shared folder is removed and the link is deactivated.

III. PHR Vault Inpelemantation

Using the application model presented above, we can implement a Personal Health Records that permit to securely store personal Health records, and allow the deposits of records and securely share records with a third party without the need to use unsecure internet. Because we can't impose on others what to use as a mean of communication over the internet, we can impose how to communicate with us when it comes to PHR.

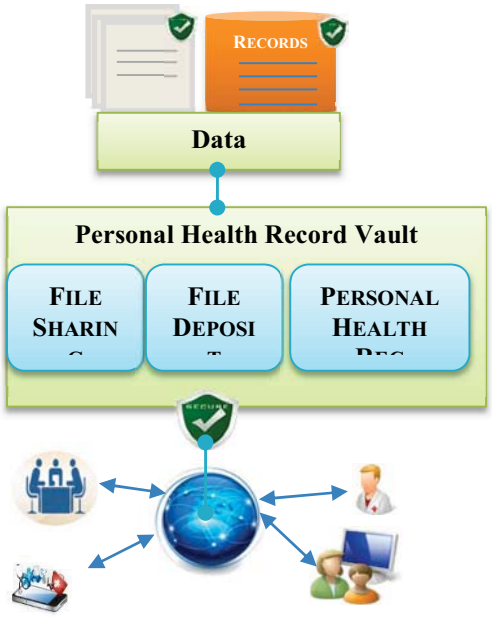


Fig 8 –Personal Health Record Implementation

A. Permanent Actors

- Data Owner is the Custodian of the Personal Medical records. He can add others to be custodians such as a caregiver or family.
- An Expert could be a user with read only rights.

All 3 authentications methods are available for them to use. We recommend that these actors to use authentication method 3: Secure access from an authorized device.

B. Manual Direct Deposit using Direct File Deposit

The custodian of the PHR sends a secure request to a provider to deposit PHR. The custodian of the records does not have to manage additional security settings for the recipient. The benefits: no integration between Custodian and Provider. The PHR Vault secures the communication. No records are transferred using emails or public shared folders. No need to sanitize the Internet.

C. Record Sharing with an ouside entity

The Custodian of the records could share his personal records with experts, lawyers, and others using the method described above, without compromising the confidentiality or the privacy of the records.

D. PHR TO EHR GATEWAYS

The power of the platform is that we can securely add more services without compromising security. The API layer could accommodate one or many PHR gateways to one or many EHR systems. The PHR Vault gateway registers with EHR for one or multiple patients. EHR sends a notification to the gateway each time there is an updated record. The Vault Gateway authenticates to EHR using proper credentials and extracts the modified records.

Benefits: PHR to EHR – Gateways:

- Providers do not need to develop or maintain patient portals. Instead of opening up its firewall to all patients to access their medical records, it is only one trusted relationship to be managed
- Patients can access their medical records at will, and can easily obtain a second opinion, without the consent of the primary provider
- Gateway also could be two-way data exchange where the PHR collects information about the patient and certain data such as remote monitoring information, or emergency treatment could be transferred back to the EHR hosts. The Provider does not need to expose their EHR system to remote monitoring devices
- Emergency Medical Attention is more efficient as the patient or a custodian could share the personal health record with the emergency provider accurately

Challenges:

"Extracting clinical information from diverse EHR systems requires extensive use of standardized vocabularies and terminologies, as well as robust information models for storing, discovering, and processing that information" [16]. However, a lot of the ground work has already been laid and working with consortiums will be a critical part to succeed.

E. IoT Devices to PHR

The biggest prize for the use of PHR is the potential integration with IoT Health devices. IoT medical device use is growing exponentially. Unfortunately, today, each time a person acquires and uses an IoT device, the generated PHR record goes to the device's associated cloud. The Custodian has little control over the process. One person may have his PHR spread all over the internet. The privacy of these records is at a best risked if the privacy was not already breached. Also, one can't reap the benefit of collecting health information as it is spread; the content is not consolidated. The challenge is to get the IoT Personal Health providers on board, as it might have negative economic implications.

IV. Conclusion

We believe that with this model, we can improve security in cloud computing and thereby increase the users' confidence in using the cloud. Our model works across various different

vertical markets. In this paper, the model was applied to Personal Health Records, while Legist LLC applied it to develop suite of applications (email, case management, add cloud File Server to share files in the cloud) targeting legal professionals. The design used by Legist is identical to the one presented in this paper. The system satisfies the privacy and confidentiality requirements for a law firm: Secure Authentications, Encrypted Documents-logs- database files, obfuscated file names, emails, and case information. -the pilot sites are pleased with the ease of use and performance of the system. Authentication method 2 is used by all pilot users while authentication method 3 is used for inbound and outbound files. Each firm has its own Storage Bucket and DB instance.

A. *Queries Testing*

Queries testing shows less than 1 sec of reasonable-overhead caused by obfuscation (1380 ms -421ms) targeting 1.2 million records.

Queries Sample	records retrieved	Time in ms No Obfuscation	Time in ms with Obfuscation	Targeted records
Fetch all files (names, source, type) with modified date in the past 30 days	3000	421	1380	1,200,000
Fetch email with keyword "barracuda"	65	3000	3310	100,000

Table 2 Query Response time – Database encrypted at rest

B. *Penetration testing*

Penetration testing was very positive. The model is secure; however, it cannot mitigate all the vulnerability caused by poorly written applications.

Tests Conducted	Tools used	Vulnerability
Authentication & session Mgt.	Selenium - Webcruiser	None – However more testing need to be done- improve dictionary for brute attacks
Security Misconfiguration	Zenmap -WebCruiser	Mitigated with proper Configuration (site owner)
Url Manipulation	Webcruiser	Require proper application discipline
Sql Injection	Webcruiser	Require proper application discipline
Cross-site scripting	Webcruiser	Mitigated by policy manager

Table 3 Penetration testing

C. *Latency Testing*

The added proxies seem to have little impact on the performance, the added latency is <3 ms
Tools used Selenium - Selenium testing machines were permitted to access the API server directly.
Sample test script was designed to call certain function directly against API followed by calling it through the reverse proxy. The script runs for 1,10,100,500,1,000,10,000 users concurrent with 100 requests loop for each user (pause 1 sec)

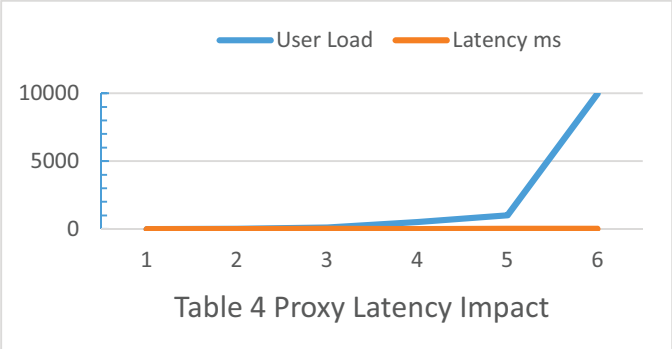


Table 4 Proxy Latency Impact

V. Future Work

Mayo Clinic: We already started "Tele-Vision" remote app, on mobile devices. The partnership will be crucial to understanding the readiness of Health Care Providers to open their system for integration with Personal Health Care records.
Intelligent Policy Manager: Work on patterns and predicability of user interaction with the system which will lead to a more secure system.

VI. Related Work

Our work was cultivation of knowledge from IBM Worklight[2],Spring Framework[17], and influenced by security principles [18]

References

[1] The Apache Software Foundation

[2] Irvine, Megan, and Jason Maddocks. *Enabling Mobile Apps with IBM Worklight Application Center*. IBM Redbooks, 2013.

[3] Song, Dawn, et al. "Cloud data protection for the masses." *Computer* 1 (2012): 39-45.

[4] Reddy, Varun, and Jagadeeshwar Rao. "DPaaS security suit for data protection in Cloud." (2014).

[5] Raizen, Helen S., et al. "Systems and methods for selective encryption of operating system metadata for host-based encryption of data at rest on a logical unit." U.S. Patent No. 8,261,068. 4 Sep. 2012.

[6] Garg, Shelly, et al. "Candidate indistinguishability obfuscation and functional encryption for all circuits." *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013.

[7] Damiani, E., et al. "Computing range queries on obfuscated data."*Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*. 2004.

[8] Naehrig, Michael, Kristin Lauter, and Vinod Vaikuntanathan. "Can homomorphic encryption be practical?." *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011.

[9] Wang, Zheng-Fei, et al. "Fast query over encrypted character data in database." *Computational and Information Science*. Springer Berlin Heidelberg, 2004. 1027-1033.

[10] Wang, Zheng-Fei, et al. "Fast query over encrypted character data in database." *Computational and Information Science*. Springer Berlin Heidelberg, 2004. 1027-1033.

[11] Li, Ming, et al. "Authorized private keyword search over encrypted data in cloud computing." *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011.

[12] Amer Jneid et al. *"SYSTEM AND METHOD FOR PROVIDING MULTI FACTOR AUTHENTICATION"* U.S. Patent Application No 14/948,124

[13] Rao, T. Venkat Narayana, and K. Vedavathi. "Authentication using mobile phone as a security token." *IJCSET* 1.9 (2011): 569-574.

[14] Praveena, T. Lakshmi, and V. Ramachandran. "Attribute based Multifactor Authentication for Cloud Applications." *International Journal of Computer Applications* 80.17 (2013)

[15] Hhs.gov, 'Summary of the HIPAA Privacy Rule', 2015. [Online]. Available:http://www.hhs.gov/ocr/privacy/hipaa/understanding/summary/index.html

[16] Pathak, Jyotishman, et al. "Normalization and standardization of electronic health records for high-throughput phenotyping: the SHARPn consortium." *Journal of the American Medical Informatics Association*

[17] Johnson, Rod, et al. *Professional Java Development with the Spring Framework*. John Wiley & Sons, 2009.

[18] Whitman, Michael, and Herbert Mattord. *Principles of information security*. Cengage Learning, 2011.

Mitigation Model to Improve Android Security

Hani Alshahrani, Ali Alshehri, Raed Alharthi and Huirong Fu
Department of Computer Science and Engineering
Oakland University
Rochester, MI USA
{hmalshahrani, aaalshehri, rsalharthi, fu}@oakland.edu

Ye Zhu
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, OH USA
y.zhu61@csuohio.edu

***Abstract*—The popularity of mobile devices that run Android Operating System (OS) has gained many attackers' attention to develop malicious applications to attack users' phone. The growing amount of these applications leads other researchers to develop new techniques to defend against these attacks. In this paper, we propose a new method, which will check any application during the installation process. This method will prevent any application that has malware from being downloaded onto the user's device.**

Keywords- Android; APK; application checker; malware.

I. INTRODUCTION

The Android OS has captured more than 83.1 percent of the total market-share comparing to other smartphone OS [1]. Eweek.com published a study citing that malware has infected 68% of the mobile devices and these data values were collected from cellular network resources [25]. The study concluded 99 percent of the infected mobile devices were, in fact, running on Android OS [25]. Webroot's Mobile Threat Report supported this result as well. This report spotlighting the four million applications, found in Google Play Market, was based on Android OS. This report indicated, 15 percent of these applications were malicious, suspicious, and unwanted. However, 14 percent were listed as trustworthy [20]. These aforementioned data values compelled researchers to yield solutions for averting these higher numbers of attacks.

Android has attracted the higher number of developers because it is an open source operating system. Developers can release their compelling products to the consumers without many restrictions through Google Market, Android's official app market, with almost no cost. Therefore, Google Market has been experiencing a spike in the number of apps downloaded. However, the consumer's security has been compromised. The less restrictions from Google Market has opened the door to all third party developers but meanwhile jeopardizing consumer's security from malware, virus, and attackers.

Hence, we proposed this model in order to improve Android security by scanning all apps before they get installed onto a device. We consider two scenarios in our proposal. First, we apply signature based detection and static analysis on the desired application which will take place online in a cloud. Second, if the Internet connection was not available during the installation process, which means the application was installed manually through USB ports or SD cards, we apply static analysis and a lightweight method analysis on the device.

The rest of this paper is organized as follows: Section II background of Android, which explains the Android architecture and pre built security features. In Section III we present the Android malware and current detection methods as well as its limitations. After that, in Section IV we introduce our model describing the approach of it followed by the related work in Section V. Lastly, in Section VI the conclusion of the paper and the future work.

II. BACKGROUND

A. Overview of Android

Android is an open source operating system based on Linux Kernel that runs on most smartphones, tablets and other devices. It is developed by Google and promoted by the Open Handset Alliance (OHA). Therefore, you can get connected to most of Google features and applications through the Android OS on your smartphone or tablet. On October 2015, Google officially released the latest version of Android mobile operating system, which is 6.0, or as they call it, Android Marshmallow [24].

B. Android Architecture and Security Features

The Android architecture is like a software stack consisting of four layers as shown in Fig. 1, to facilitate the running of applications. The base layer is the Linux Kernel, which has the essential hardware drivers. The next layer right above the kernel is the native libraries and the Android runtime. The native libraries are written in (C\C++) and enable the device to handle different types of data. The Android runtime consists of Android Virtual Machine (Dalvik) and a set of libraries that offers most of the functionality of the Java core libraries. The Dalvik VM was developed by Google and relies on the underlying Linux Kernel for low-level functionality [2].

The next layer is the Application Framework, which developers directly interact with to build applications. Android framework contains all key managers such as activity manager, package manager, location manager and telephony manager. The very top layer in the Android architecture is the application

layer, which includes the pre-installed applications and the applications installed by the user [3].

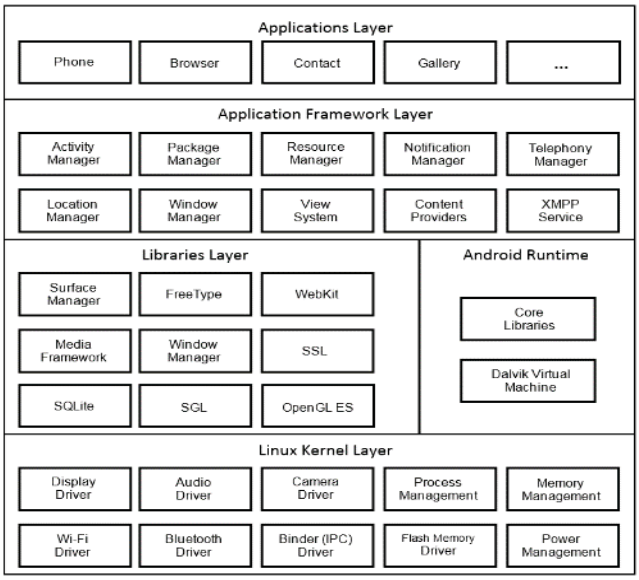


Figure 1. Android architecture.

There are some security instruments pre built into the Android operating system to protect data and to reduce the effect of application security problems. This allows the developers to build their applications with default system and file permissions and without worrying about difficult decisions regarding security [5]. We discuss some fundamental security mechanisms below, which include sandbox, permission-based access control, data encryption, safe memory management, and secure Inter Process Communication (IPC).

- *Sandbox*: It provides an isolated environment to execute downloaded applications on Android, so every application operates in separate virtual machines. It also controls the resources, thus the data, memory and code of an application cannot be accessed by any other application.
- *Permission-Based Access Control*: In the Android system, user-granted permissions restrict access to system features and user's data. Because each Android application runs in a separate sandbox, applications need permissions to share data and resources with other applications. Each application must state the required permissions during the installation process and the user has to give or deny the permissions. However, if the user chooses to deny the permissions the application will not be installed [6].
- *Data Encryption*: Android OS users have a great built-in security feature, which is the ability to encrypt their personal information and data such as accounts, settings, phone numbers, passwords, downloaded apps, media and files. Encrypting the data in the Android system is more than simply setting up a lock screen code; it is about providing the users with a protection level to save them from any attack. So, all the data on your device is encrypted with a special cryptographic key. On the other hand, if you want to decrypt your device you need to enter a password each time you power it on. The encryption process costs time because it takes an hour or more. Also, it consumes power because the device must be connected to the power until the encryption is completed; otherwise there is a chance of losing some of your important data [7].
- *Safe Memory Management*: Android has its own runtime and virtual machine (Dalvik), which is used to manage application memory. Unlike other frameworks, the Android runtime also manages the process lifetimes. Android OS ensures application responsiveness by stopping and killing processes as necessary to free memory.
- *Secure Inter Process Communication (IPC)*: This feature is supported through the binder framework in Android and allows two applications to communicate. Because one application cannot access the other application's data and memory directly, the IPC is responsible to handle the communication between them [8].

C. Installation Process of an Android Application

Before explaining the installation process, we will discover two important default applications for Android that are responsible for installing, uninstalling, and upgrading any package in an Android system. The first application is Package Manager, which is an application program interface that manages all packages' installation, uninstallation and upgrades. Package Manager is responsible to get the confirmation from the user before installing any package in the system. The second application is *PackageInstaller* that provides a user with an interface in order to manage all packages and applications [9].

An Android app is saved into an APK (.apk) zip file and it can be installed from the Android market into the user's device. This zip file contains some files such as *AndroidManifest*, *lib*, *classes*, and *recourses*. Also, it contains several folders like *Meta-Inf* and res. In this section, we will explore each file and folder in an APK zip file and illustrate the importance of it. As shown in Fig. 2, the APK file contains some folders and files, and the most important file in the package is AndroidManifest.xml. This file contains all critical information about the app, which will be presented to the Android system. That information must be provided to the system before any code of the app can run [11].

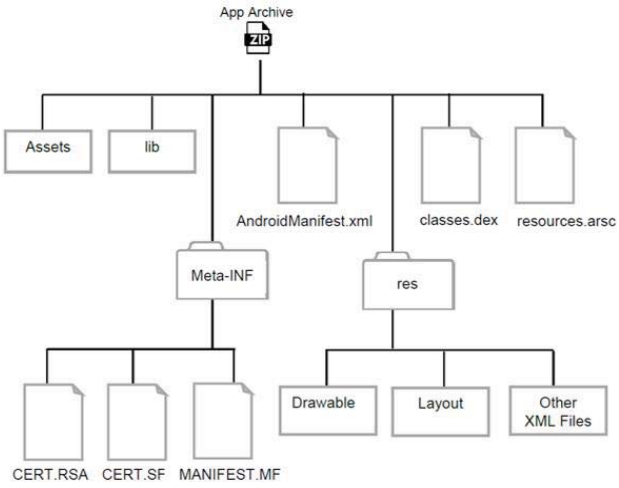


Figure 2. APK structure.

Assets file is used to save raw files. Libs file contains all private libraries. Res file contains all recourses for the application such as layout files. *META-INF* saves all developer's information, such as signature and certificate to confirm the third party developer identity. *Classes.dex* contains all classes that are compiled to the *.dex* format to be executable by the Dalvik virtual machine. *resources.arsc* is a file that contains all resources that already compiled such as binary XML [12].

All Android applications are developed in the Java language, and are then compiled to Dalvik bytecode to be executable by the Dalvik Virtual Machine. The developer can compile Java code to Java-bytecode by creating numbers of *.class* files. By using *dx* tool the *.class* files will combine into one Dalvik Executable (*.dex*), which saves Dalvik bytecode in order to execute it on the Dalvik Virtual Machine [10]. Fig. 3, illustrates the development process.

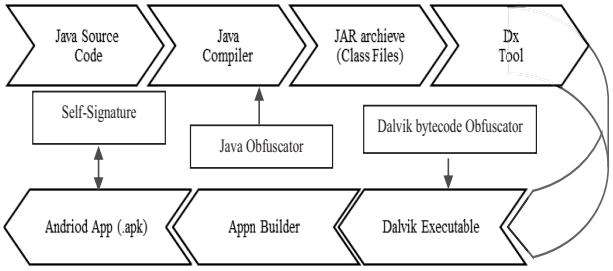


Figure 3. Development process.

D. Application Components

Android app contains four types of app components (activities, services, content providers and broadcast receivers) that allow the system to communicate with the app.

- *Activities*: This component is the user interface of the application. It can be defined in the *AndroidManifest* file and it could be the entry point to the application. An application frequently has multiple activities. As such, an email app has an activity to display all new emails, another activity to read emails, and another activity to send a new email. Any one of these activities can be started by different apps once it has the permission. It also can be started for returning results to its caller [13].
- *Services*: This component runs in the background to perform work for remote processes or to perform operations for long running apps without providing a user interface. For example, playing music in the background, and fetching data from the Internet while the user is on a different application [14].
- *Content Providers*: This component manages access to a structured set of data. Also, it provides security mechanisms for data. Applications can share data or even alter it (if the app has the permission) via the content provider. For example, any app with the appropriate permissions can edit or read user's contact information. In addition to that, content providers can connect code running in one process with data in another process [15].
- *Broadcast Receivers*: This component is responsible for listening to the Android OS events. For example, it can initiate a broadcast announcing that the battery is low. Furthermore, apps can start broadcasts such as informing other apps that some data has been installed to the device and they can start using it [16].

III. Android Malware and Detection Methods

Nowadays, Android malware has been a major security concern for Android users. Malware is a type of software that runs to threaten, attack, and steal user's private information, or expense their account without the user's agreement or prompt. It can be hidden in the device memory without the user's knowledge and spy on the browser and system activities.

A. Type of Malware

In our paper we look over common existing Android malware. Also, we mention what kind of damage that might cause.

- *Virus*: Viruses are self- multiplying programs that emerge in mobile smartphones to delete or convert user's data on the handset. They are usually disguised as a game, a security patch, or other attractive application, and are then installed into a smartphone.
- *Worm*: Worms are malicious programs which have the ability to replicate itself by executing its own code depending on any other program. The purpose of a worm is to occupy both system and network resources. The main difference between viruses and worms is the method in which they reproduce and spread.
- *Spyware*: Spyware is a program that collects user's information and smartphone data without users' knowledge or permission. In addition, it may control the smartphone without the user's consent. Spyware can be used for tracking and storing Internet user's movements on the Internet and serving up pop-up ads to the Internet user.
- *Botnet*: Botnet is a program which enables an attacker to control the infected device with zombie programs to perform malicious tasks at the same time. For instance, sending lots of spam texts or emails to a specific destination.
- *Trojan*: A Trojan is another type of malware that can be installed in the device causing loss of data. Most Trojans in smartphones are related to activities such as recording calls, instant messages, specific location via GPS, forwarding call logs and other vital data. This type of malware runs in the background of an application and does its attack without the user's attention. It can increase the phone bill by sending SMS to premium mobiles and also blocks messages from service providers to users alerting them of additional charges.
- *Backdoor*: Backdoors are malware program (cryptosystem or algorithm) methods that allows other malware to discreetly enter the system by passing normal authentication and obtaining access to plaintext. Backdoors can exploit root to access the super-user privilege in order to hide from anti-malware programs.

B. Current Detection Methods and Limitations

In this section we describe the current methods used for malware detection, and then we give the limitations of each technique.

1) Anti-Virus: According to Fedler et al. (2013) [17], there is a number of antivirus software existing for users, which can scan any app for malware. However, Antivirus software does not have the ability to access or monitor an Android device's file system or dynamic behavior of installed apps. Thus, malicious files can still download without being openly code and executed in the device without being detected by antivirus software [17]. Anti-virus software cannot access any other apps data unless it's included in the anti-virus database [17]. Furthermore, Anti-virus software scans the application for malware after downloading the application on the device [18].

2) Google's Bouncer: Bouncer was rolled out in February 2012 by Google to filter out malicious apps before they ever uploaded in the Android Market (Google Play). The approach of Bouncer is protecting Android from worms and Trojan horses. Bouncer had been quietly running for several months, the result was a 40% drop in potentially malicious apps in the Market. However, Bouncer can be broken when a malware author knows its inner workings. A malware author could build a module that suspends malicious behavior for a certain amount of time when Bouncer is detected. Bouncer doesn't run apps continually; in fact, it will scan each app only while uploading and declaring the result. The malware developer just needs to keep their intentions hidden for a short time to avoid the scanner as it exists now. On the other hand, malware authors could avoid detection by playing it cool [19].

IV. Proposed Model

Google released Android's source code under the Apache license. Thus, users can modify the software freely using this permissive license [18]. In our proposed system we are trying to develop a third party application called (Application Checker), that is going to check all applications during the installation process from any malware or over privilege permissions that some apps require in order to run third party libraries such as advertisements. Hence, to run our application we have to hook our application with the package manager, which requires system permission. Fig. 4, illustrates our proposed model.

To achieve our goal, we consider two scenarios when users want to download any application into their devices. First, when an application is downloaded into the device using USB or SD card, we called this scenario "Offline Downloading". Second, when an application is downloaded from Google Paly or third- party markets using the Internet connection, we called this scenario "Online Downloading".

A. Offline Downloading

In our Android OS allows users to download apps from their own computers using USB ports after they downloaded the preferred app from the Internet. Also, some merchants sell some SD cards that already have some APK files installed on it. If the user wants to install any app he/she can just run the APK file that's already installed in order to run the application in the device.

The application checker is going to apply two procedures. First, the application checker will apply static analysis in order to gather all libraries that have been used by the chosen application. For this reason, it is going to evaluate the *AndroidManifest* file of the application to determine if the developers isolate the advertisement's code from the actual app's code as Zhang et al. (2013) proposed [21]. Second, the application checker will apply a lightweight method analysis that will detect malware on the device in a reasonable time; this method has been proposed by Arp et al. (2014) [22].

1) Static Analysis of AndroidManifest:

In this step we focus on how to complete the code analysis in a short time period since users will not wait a long time in order to get the result. We decided to analyze the *AndroidManifest* file since it has enough information that will

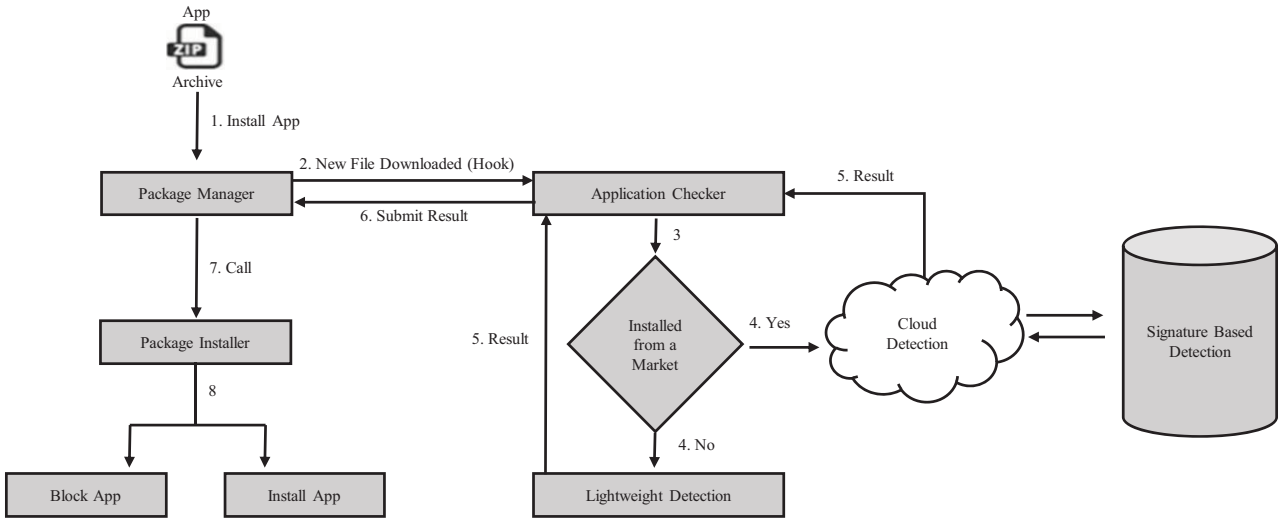


Figure 4. Block diagram of proposed model

allow us to know whether the developer did isolate the advertisement's code from the actual code or not. Hardware components, requested permissions, app components, and filtered intents are examples of some information that we can get from *AndroidManifest* file.

The application will check for all activities in *AndroidManifest* file. According to Zhang et al. (2013) [21], isolating third-party code from the actual code will restrict the over privilege permissions that will be given to the desired app. Since adware is the most common malware that affects most users' devices and it's spread through advertisement libraries. The application checker will analyze the *AndroidManifestfile* to see if the advertisement activity is run in separate activity and if it has its own permissions. If the application checker finds that the advertisement activity is not isolated from the main activity, then a warning message will be displayed to the user. As reported by Zhang et al. (2013) [21], they developed a method called Aframe that will isolate permission required by any third-party library as well as isolate the display and the input of that library.

2) Static Analysis of classes.dex file

After analyzing the *AndroidManifest* file we can also analyze the *classes.dex* file, which is containing the source code of the application. Analyzing the *classes.dex* file will provide our application checker with some information such as restricted API calls, used permissions, suspicious API calls, and network addresses.

Most malicious activity is showed in particular patterns and mixture of the extracted information. For example, a malware sending SMS will need for permission SEND_SMS and hardware component to send the message *android.hardware.telephony* [22]. Create Boolean expressions will catch these dependencies between extracted information from *AndroidManifest* activity file and *classes.dex* file and return true if malware is detected [22]. Therefore, the application checker will return a message to the package manager in order to block the installation and provide the user with a warning message about the found malware. Otherwise, the application checker will return a result that shows no malware was found in the desired application.

B. Online Downloading

As we mentioned before what we mean by online downloading is any application that is installed in the device using Internet connection. In this scenario we aimed to use the signature based detection approach to identify and detect any malware. However, we cannot have the signature based detection method working on-device because it requires frequent updates of the virus signature dictionary, which it is difficult for the device to handle, due to the limitations of it such as storage, processing and power. Therefore, the user will not be satisfied if the malware detection approach slows down the OS performance or drains the battery quickly. Consequently, it is better in this case to have the signature based detection approach working on the cloud because of two reasons. First, any new virus signature will be updated to the virus signature dictionary without affecting the users' data plan. Second, using the cloud will not affect the mobile device's storage and power.

If an application was downloaded from Google Paly or third-party markets using the Internet connection the application checker is going to send the *AndroidManifest* file of the desired application to the cloud in order to apply two procedures on it. First, apply signature based detection method to check if there is any malware on the application or not. The reason of using *AndroidManifest* file is to analyze and discover any virus signature (viral code) in it. If a virus signature is found in the file, the application checker will inform the package manager that malware was found to take action and block the installation process. Otherwise, the application checker will return a result that shows no malware was found in the desired application. Second, apply static analysis in order to confirm if the third party libraries are separate from the actual app's code or not (as we already described in the offline downloading).

V. Related Work

The detection and mitigation of Android malware has been an interested area of research in the last few years. Numerous thoughts and techniques have been proposed to overcome the growing amount of malware. Table I shows some recent research on Android security. The table presents each work with its main goal. We built our proposed model based on some theories that were proposed with the aim of mitigating some risks in Android OS.

Zhang et al. (2013) [21], proposed AFrame which will run the advertisement and the application in separate frame. It provides isolation for both input and display. Therefore, our model will check if the developers implement Aframe approach in their applications or not. Also, Arp et al. (2014) [22] introduced Dreben as a lightweight method for detection of Android malware that enables identifying malicious applications directly on the smartphone. However, this detection method cannot detect malware or malicious applications before the installation process. However, the main goal of our proposed method is to detect any malicious applications before they installed. Pearce et al. (2012) [23] proposed AdDroid which separates privileged advertising functionality from host applications to allow applications to show advertisements without requesting privacy sensitive permissions. Thus, in our model we want to inform the user if the application contains any advertising library that is not separated from the host application. Nevertheless, Bhonde and Chatterjee [18], proposed a model to check application before instillation. However, their model applies checksum using secured hash algorithm on the installed app to check if it has a malicious signature or not. Therefore, our model will apply dynamic analysis and static analysis to get more accurate results about the installed application.

TABLE I. Research on Android Security

Previous work	Goal of the research work
Aframe [21]	Isolate third-party code from the actual application's code.
DREBIN [22]	Detect malware using lightweight method.
Android Application Assessment [18]	Check application before installation.
AdDroid [23]	Separate Ads library from host app.

VI. Conclusion and Future Work

Among different smartphones' OS, Android OS is the main target for most of the malware developers, because it is an open source with less developing restrictions. A large number of users download apps from the Google market every day. Consequently, our desired impact in this paper is to assist

Android OS users in detecting the malware before it is installed on their devices. In this model, we applied the signature-based detection approach in the cloud to avoid the limitations of mobile devices. Also, a lightweight method analysis is used in case the app was not installed from the Google market. This process does not consume any extra power because it only works during the installation process to increase the efficiency of the mobile device. Overall, mitigating the malware attacks will improve the Android OS security and efficiency. Implementing the proposed method in the real world is our future work with this mitigation. We will evaluate the model's performance by applying it in many different malware families.

REFERENCES

[1] "Gartner Says Sales of Smartphones Grew 20 Percent in Third Quarter of 2014." *Gartner*. [Online]. Available: http://www.gartner.com/newsroom/id/2944819. Accessed on: 18 Feb. 2015.

[2] C. Wang, W. Duan, J. Ma and C. Wang. "The Research of Android System Architecture and Application Programming," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference,* 2011, pp. 785-90.

[3] "An Overview of the Android Architecture." *Techotopia*. [Online]. Available:http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture. Accessed on: 01 Mar. 2015.

[4] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer. "Google Android: A Comprehensive Security Assessment." *Security & Privacy, IEEE,* vol.8, no.2 , pp. 35-44, March-April 2010.

[5] V. Cooper, H. Shahriar, and H. Haddad. "A Survey of Android Malware Characterisitics and Mitigation Techniques." *Information Technology: New Generations (ITNG), 2014 11th International Conference,* 2014, pp. 327-332.

[6] "Security Tips". *Android Developers*. [Online]. Available: http://developer.android.com/training/articles/security-tips.html. Accessed on: Feb. 2015.

[7] Martin Brinkmann. "Encrypt all data on your Android phone". *Ghacks.net*. [Online]. Available: http://www.ghacks.net/2012/10/13/encrypt-all-data-on-your-android-phone/. Accessed on: 15 Feb. 2015.

[8] "Android AIDL Example with Code Description – IPC". *Techblogon*. [Online]. Available: http://techblogon.com/android-aidl-example-with-code-description-ipc/. Accessed on: 20 Feb. 2015.

[9] K. Parmar, "In Depth: Android Package Manager and Package Installer". [Online] Available: http://java.dzone.com/articles/depth-android-package-manager. Accessed on: 15 Mar. 2015.

[10] P. Faruki et al. "Android Security: A Survey Of Issues, Malware Penetration And Defenses," in *Communications Surveys & Tutorials, IEEE*, vol.17, no.2, pp.998-1022. 2015.

[11] "Fundamentals". *Android Developers*. [Online] Available: https://developer.android.com/guide/topics/manifest/manifest-intro.html. Accessed on: Mar. 2015.

[12] "Managing Projects Overview". *Android Developers*. [Online] Available: https://developer.android.com/tools/projects/index.html. Accessed on: 15 Feb. 2015.

[13] "Activities". *Android Developers*. [Online] Available: http://developer.android.com/guide/components/activities.html. Accessed on: 01 Feb. 2015.

[14] "Services." *Android Developers*. [Online] Available: http://developer.android.com/guide/components/services.html. Accessed on: 01 Feb. 2015.

[15] "Content Providers." *Android Developers*. [Online] Available: http://developer.android.com/guide/topics/providers/content-providers.html. Accessed on: 01 Feb. 2015.

[16] "Fundamentals". *Android Developers*. [Online] Available: http://developer.android.com/guide/components/fundamentals.html. Accessed on: 01 Feb. 2015.

[17] R. Fedler, M. Kulicke, and J. Schutte. "An Antivirus API for Android Malware Recognition," in *Malicious and Unwanted Software: "The Americas" (MALWARE), 2013 8th International Conference,* 2013, pp.77-84.

[18] A. Bhonde and M. Chatterjee. "Security Solution for Android Application Assessment, " in *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 3, no. 9, pp. 7987-7991. 2014.

[19] R. Whitwam. "Circumventing Google's Bouncer, Android's Anti-malware System". *ExtremeTech*. [online] Available: http://www.extremetech.com/computing/130424-circumventing-googles-bouncer-androids-anti-malware-system. Accessed on: 15 Mar. 2015.

[20] "Malicious Adware Continues To Plague Android Apps". *Prosecurityzone*. [online] Available: http://www.prosecurityzone.com/News_Detail_Malicious_adware_continues_to_plague_android_apps_22969.asp#axzz3XWc7MRgK. Accessed on: 15 Mar. 2015.

[21] X. Zhang, A. Ahlawat, and W. Du. "AFrame: Isolating Advertisements from Mobile Applications in Android," in *the 29th Annual Computer Security Applications Conference (ACSAC '13), 2013, pp. 9-18.*

[22] D. Arp, , M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck. "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket," in *the Annual Symposium on Network and Distributed System Securit,* 2014.

[23] P. Pearce, A. Felt, G. Nunez and D. Wagner. "Addroid: Privilege separation for applications and advertisers in android," in *the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS '12)*, 2012, pp.71-72.

[24] "Get ready for the sweet taste of Android 6.0 Marshmallow". *Official Android Blog*. [Online]. Available: http://developer.android.com/training/articles/security-tips.html. Accessed on: Oct. 2015.

[25] R. Lemos. "Mobile Malware Mostly Infecting Android Devices Rises Steadily." *EWeek*. [Online] Available: http://www.eweek.com/security/mobile-malware-mostly-infecting-android-devices-rises-steadily.html. Accessed on: 30 Mar. 2015.

Live Migration of Virtual Machine in Cloud: Survey of Issues and Solutions

Hani Alshahrani, Ali Alshehri, Raed Alharthi, Abdulrahman Alzahrani, Debatosh Debnath and Huirong Fu
Department of Computer Science and Engineering
Oakland University
Rochester, MI 48309, USA
{hmalshahrani, aaalshehri, rsalharthi, aalzahrani, debnath, fu}@oakland.edu

Abstract— **Cloud computing is a technology that is built on the theory of virtualization- the idea to allow multiple virtual machines to share the resources of a host. One of the features of virtualization is live migration. This feature will allow one or more virtual machines to be migrated from one host to another without interfering with applications running on those virtual machines. However, there are many security issues associated with virtual machine live migration, which discourages many IT managers of taking advantage of this technology. This survey discusses live migration security issues, along with possible attacks before, during, and after migration. In addition, this survey discovers some known methodologies that aim to secure virtual machine live migration.**

Keywords—component; VM; virtualization; VMM; cloud computing; live migration security.

I. INTRODUCTION

Cloud computing helps small and medium companies by providing infrastructures at a low cost. Also, it allows individual users to have access to resources as long as they have connected devices. The idea of cloud computing is to move computing desktops, such as computer processing, storage applications and services, to a service-oriented platform which can be achieved by using both a server cluster and a huge database. Cloud computing comes in three types, public, private, and hybrid, which have four design goals: scalability, virtualization, efficiency, and reliability [1]. In order to offer application flexibility, the public cloud can be accessed by any public user who has already subscribed to the service. The private cloud is maintained by a single organization and is accessed by limited users in order to have higher security and efficiency. The hybrid cloud leverages the benefits of both public and private clouds to provide better service for the users. Moreover, cloud computing provides three type of services: infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS).

Cloud computing was developed from some other technologies such as cluster, grid, virtualization, Service Oriented Architecture, and web 2.0 services [2]. The security issues of those technologies were therefore inherited by the cloud. This is the main reason that many IT managers hesitate to adopt the cloud. Virtualization is the main component in the IaaS model. It virtualizes all resources (e.g., servers, storage, networks, and data center fabric) requested by the users in the IaaS model in the form of a virtual machine (VM). Using VM has many benefits, such as enabling resource sharing and easily allowing the transfer of different VMs between physical hosts. However, VM has major vulnerabilities and threats, such as VM poaching, VM jumping, and unsecured live VM migration [2].

Live migration of VM allows transfer of one running VM from one physical server to another without interrupting the services that are currently running in VM. For instance, some cloud computing providers have hundreds or even thousands of VMs that are spread throughout a number of servers. For any reason (e.g., fault takeover, workload balancing, and hardware maintenance), the provider might migrate one VM from one physical server to another [3]. This migration might cause several security risks in both data centers and cloud computing. One example of a security risk that might be caused by the VM migration is sniffing guest OS's sensitive information during the migration of VM [2]. Thus, the integrity and confidentiality of VM migrating data is compromised.

The rest of the paper is organized as follows: section II will discuss some previous work in VM live migration. Section III will give a brief background about virtualization, procedures of VM live migration, and VM live migration approaches. Threats in live VM migration and security requirements in VM live migration will be discussed in sections IV and V, respectively. Section VI will explore some of the existing solutions focusing on VM live migration security. Finally, section VII concludes the paper and discusses future work.

II. RELATED WORK

Several publications have discussed the security of VM live migration. This concept was proposed by Clark et al in 2005 [4]. In addition, Jon et al [11] explored the security issues in live migration by demonstrating the different possible attacks, such as active attack and passive attack. Moreover, Naveed et al [2] discussed the vulnerabilities in live migration as well as the possible attacks. Sebastian et al [6] developed Live Migration Defense Framework (LMDF), which can detect live migration from an affected VM. This framework will work from the inside of the VM. Nevertheless, Anala et al [3] designed a security framework to secure the live migration of VM. The framework includes defining a role-based access

policy, network intrusion, firewall protection, and encrypting the channel to achieve the confidentiality of migrating data over the network.

III. BACKGROUND

In this section, virtualization technology and VM migration will be discussed, including its types and objectives. Also, the procedures of VM live migration, along with its performance measurements, will be explained.

A. Virtualization Technology

In 1960, virtualization technology was implemented on the IBM mainframe [7]. Virtualization is a computer architecture technology where multiple VMs share the same hardware of the machine. The objective of using a VM is to enhance the sharing of resources (e.g., storage, network, CPU, etc.) among many users and to improve the performance of a computer through utilization of resources and the flexibility of applications [1]. The main idea of cloud computing is virtualization. Storage, applications, etc. can be virtualized in cloud computing. Virtualization of the host operating system's physical hardware run by the VMs is done by a software layer called a virtualization layer. This layer is known as hypervisor or virtual machine monitor (VMM). Xen [8] and VMware [9] are examples of virtualization software.

B. VM migration types and objectives

As mentioned before, virtualization is the main element in today's cloud computing. It separates the physical hardware from the guest operating system (OS). This separation allows the guest operating system to migrate from one physical server to another. There are two types of VM migrations: non-live migration or live migration. In non-live migration, all applications running on the VM will be stopped during the VM migration, while in live migration, all applications continue running without any interruption. Moreover, there are several objectives of VM migration, such as power management, load balancing, and system maintenance [5]. Power management will power off VMs in underutilized servers in order to ensure power saving. Moreover, load balancing will help to avoid overlap by migrating VMs from a host with a heavy load to another host with a lesser load. Additionally, system maintenance will improve the reliability and availability of the system.

C. Procedures of VM live migration

The actual migration process includes several stages [4]. Figure 1 shows the overall procedures:

1. **Pre-Migration stage:** Select both the active running VM, which will be migrating, and the receiver host along with all resources needed.
2. **Reservation stage:** Check that all resources needed are available in the receiver host, then initialize a new container for the VM of its size. If the VM successfully migrates to the new host, the memory will be copied to the new host; otherwise, the VM will continue running in the original host.
3. **Iterative Pre-copy stage:** After copying the memory for the first time in the previous stage, the virtualization layer will check the VM memory for any modification since the last copy. The VM, which will be live migrated soon, is running at the original host and its memory is modified based on its current tasks. All pages that are uncopied (called "dirty pages") will be transferred to the new host and the virtualization layer will re-check for any new modifications in the memory.
4. **Stop-and-copy stage:** In this stage, there is an interruption in about 5 milliseconds after the previous stage [6]. This interruption is necessary to move all memory pages of the VM from the original host to the new host. By the end of this stage there are two copies of the VM memory in the two hosts. In case of failure, the copy at the original host will be resumed [4].

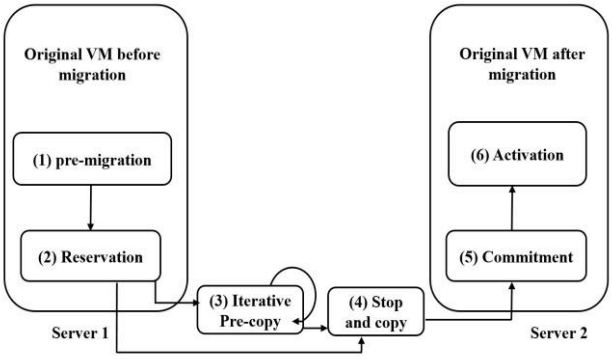


Fig. 1. Procedures of VM live migration.

5. **Commitment stage:** The original host will receive an acknowledgement from the new host, indicating that the VM is successfully migrated. Thus, the original host will release all resources used by the migrated VM and remove the original VM.
6. **Activation stage:** The VM will run now from the new host. The hypervisor will organize the network management and the VM will not change its IP address.

D. VM live migration approaches

There are two popular approaches used for memory migration in VM live migration: (1) Post-Copy approach and (2) Pre-Copy approach. First, the Post-Copy approach has two phases: a pull-phase and a stop-and-copy phase [10]. In this approach, the migrating VM will be postponed at the source. Then it will start copying processor states to the receiver host. Finally, it restarts the VM and starts transferring memory pages through the network from the source host [7]. The main advantage of this approach is that all memory pages are transferred merely one at a time. On the other hand, its downtime is higher compared to the Pre-Copy approach.

Second, the Pre-Copy approach has two phases: a push-phase and a stop-and-copy phase [10]. In this approach, all memory pages are transferred from the source host to the target host all at once without suspending the VM; this can be

achieved by the hypervisor [7]. During this phase, there might be some dirty pages. Those pages will be re-copied until there are no dirty pages at the source host's memory. Next, the VM will be suspended at the source host and restarted at the receiver host. In this approach, the downtime is fast. However, there is an overhead, resulting with some pages copied more than once during the transfer [7].

E. Performance measurements

The performance of live migration is often measured by the following measurement standards [7]:

1) Preparation Time: When the VM migration is started and the state of the VM has been transferred.

2) Downtime: The period of time where one or more services are unavailable during VM migration for users.

3) Pages Transferred: The total of all memory pages that have been transferred.

4) Total Migration: The total time of the migration, from start to finish.

5) Application Degradation: When the migration decreases the executing speed of an application in the VM.

IV. THREATS IN LIVE VM MIGRATION

Live migration is quite a new idea and its security aspects are not fully discovered. The popularity of cloud computing caught the attention of many predators, allowing them to find new ways to attack either cloud service providers or customer's data. Using live migration in cloud computing might lead to many attacks, such as denial-of-service attack, man-in-the-middle attack, etc. The existence of such issues in live migration security discourages many sectors, such as financial organizations, hospitals, and government agencies, from taking advantage of VM live migration.

Anala et al [3] and Jon et al [11] demonstrated live migration threats. Based on their demonstrations, live migration attacks can be targeting one of the three different classes: (1) control plane, (2) data plane, and (3) migration module. Figure 2 shows possible attacks during live migration.

A. Control plane

Server operations on both sides (source and receiver) are managed by a system administrator who is authorized to manage all operations through an interface console. This console will allow the administrator to perform many operations (e.g., creating new VM, migrating VM, terminate a running VM, defining the VM's setting, etc.). Hence, a lack of keeping this interface secure could be a starting point for any predator to attack the VM. Therefore, the access control of the administrator's interface must be secure and the mechanisms of communication used by the hypervisor should be authenticated and resistant against any tampering [11].

A lack of security in the control plane may allow an attacker to exploit live migration operation in different ways:

1) Denial-of-service attack: the attacker will create many VMs on the host OS for no reason other than to make the host OS overloaded, which will make the host OS not accepting of any migrated VM.

2) Unnecessary migration of VM: In this situation the attacker will overload the host OS by unneeded VMs. This will run the dynamic load balancing feature. This feature will migrate some VMs from a loaded host to another unloaded host.

3) Disrupt the regular operations of the VM: An attacker may migrate a VM from one host to another host without any goal except to interrupt the operations of the VM.

4) Attack on VMM and VM: In this kind of attack the predator will migrate a VM that has a malicious code to a host server that has the target VM. This code will exchange information with the VMM and the target VM through a covert-channel. This channel will compromise the confidentiality of the host server by leaking target VMs' information.

5) Control of incoming migration: The attacker may migrate the target VM from one host server to the attacker host server, which results in getting full control of the target VM.

6) Advertising for false resource: This occurs when the attacker advertises false resource availability for the target VM. For example, advertising that there is a large number of unused CPU cycles. This results in migrating the VM to a compromised hypervisor.

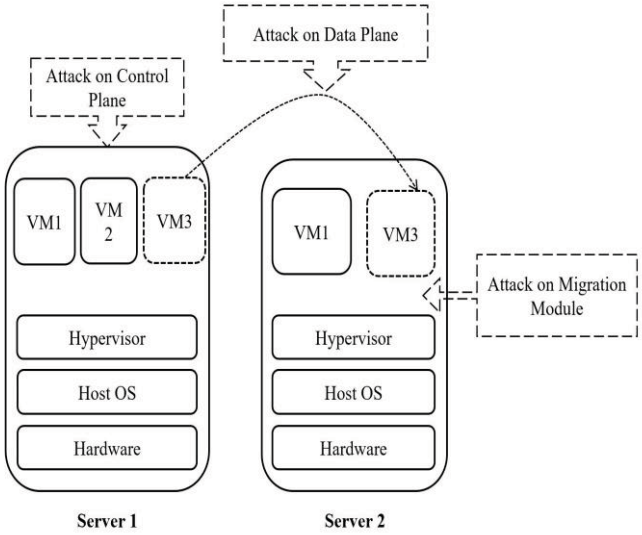


Fig.2. Possible attacks during live migration

B. Data plane

In this plane, several contents of memory are transferred from one host to another host (e.g., kernel states and application data). Thus, the transmission channel must be secured and protected against any attack. In the VM migration protocol, all migrated data are transferred as clear data without any encryption. Hence, an attacker may place himself in the transmission channel to perform a man-in-the-middle attack using one of the following techniques: Address Resolution Protocol (ARP) spoofing, DNS poisoning, or route hijacking

[11]. One of the following types of attacks can be performed through the man-in-the-middle attack: passive attack and active attack.

1) Passive attack: Observing the transmission channel and other network streams used to migrate one VM from one host to another. The attacker will be able to gain information from the VM's migrating memory (e.g., passwords, keys, application data, capturing packets that are already authenticated, messages that have sensitive data will be overheard, etc.) [3].

2) Active attack: This type of man-in-the-middle attack is the most serious attack. The attacker manipulates the migrated VM's memory when it's transferred over the network. Also, the attacker can manipulate some applications specifically inside migrated VM's memory (e.g., authentication service and pluggable authentication module in live migration) [2]. Figure 3 explains the man-in-the-middle attack during the live migration of the VM.

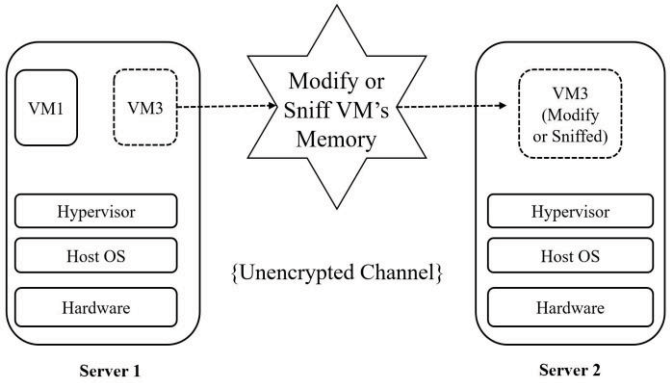


Fig. 3. *Man-in-the-middle attack during live migration of VM*

C. Migration module

Migration module is a software in the VMM that allows VM live migration. A guest OS can communicate with the host system and vice versa. Moreover, the host system has full control over all VMs running over its VMM. If the predator is able to attack the VMM via its migration module, then all guest VMs that are running above that compromised VMM will be affected in their integrity. Nevertheless, any VM in the future that has migrated to the affected VMM will be compromised, as well.

Exploiting the VM with a low security level is one of the attack techniques in the migration module. During the migration process, when attackers discover a VM with a low security level, they might be able to compromise it easily. Thus, they can use it as a gate to compromise other VMs on the same host with higher levels of security [12]. Moreover, the attacker will be able to attack the VMM by itself after identifying a way to enter the system.

V. Security Requirements in VM Live Migration

There are some security requirements that must be implemented in the VM live migration. These security requirements will enhance the security level in the previous classes to protect both VMs and host servers from any attack (before, during, and after) the live migration process. Aiash et al [13] and John et al [11] discussed security requirements in VM live migration. Following are the security requirements that should be implemented in VM live migration: (1) defining access control policies, (2) authentication between sender host and receiver host, (3) non-repudiation by source or receiver, (4) data confidentiality while migrating a VM, (5) data confidentiality before and after migration, and (6) data integrity and availability.

In this section, each security requirement will be discussed based on the classes that should be implemented. Table I shows where each requirement should be implemented.

A. Security requirements to mitigate attacks in the control plane and the data plane

1) Defining access control policies: By defining control policies on the control plane, VMs and the host server will be protected from unauthorized users. As discussed before, if attackers compromised the interface console they might perform unauthorized activities such as migrating a VM from one host to a legitimate target VMM [13].

2) Authentication between sender host and receiver host: Implement strong procedures of authentication and identification in order to prevent unauthorized users from entering administrators' interface.

3) Data integrity and availability: This requirement will stop some attacks, such as a denial-of-service attack, which causes the unavailability of either the source host or the receiver host. This can be done by applying strict policies for accessing control.

4) Data confidentiality during migrating the VM: In order to prevent a man-in-the-middle attack from getting any sensitive information, all data during migration must be encrypted.

B. Security requirements to mitigate attacks on the migration module

1) Authentication between sender host and receiver host: There should be a mechanism of authentication between the sender host and receiver host. This mechanism could involve a firewall for more security options [13].

2) Non-repudiation by source or receiver: Both the source host and destination host must observe the system's activities. All actions during live migration must be recorded [13].

3) Data confidentiality before and after migration: Data should be encrypted in both hosts' storage. If an attack happens, neither guests' VMs' data nor the host's data is affected.

4) Data integrity and availability: Virtualization software must be updated to be protected from some vulnerabilities, such as stack overflow and heap overflow [13].

VI. Existing Solutions in Live Migration Security

This section will discuss the existing solutions to secure the VM live migration.

TABLE I. WHERE EACH REQUIREMENT SHOULD BE IMPLEMENTED

Requirements Security / Live migration security class	Defining access control policies	Authentication between sender host and receiver host	Non-repudiation by source or receiver	Data confidentiality while migrating VM	Data confidentiality before and after migration	Data integrity and availability
control plane	✓	✓				✓
data plane	✓	✓		✓		✓
migration module	✓		✓		✓	✓

A. Secure Live Virtual Machine Migration (SLVM)

SLVM [3] is a role-based access control framework. This means the duty of each user on the VM should be clear and separate from other users. It actually identifies what is the operation of each administrator in the system. It has been implemented in a local area network and is not supported for wide area network.

B. Migration network isolation (VLAN)

Shetty et al [14] discussed this approach. In this approach, the migration traffic will be separate from other network traffic. Both VMs in the source and receiver hosts will be grouped together into Virtual LAN (VLAN). VLAN will allow the migration to take place in a separate traffic and the migration will be assigned into a secure transmission channel. There is one disadvantage of using VLAN, which is the increase in complexity and cost of administration when the number of VMs are increased.

C. Network Security Engine-Hypervisor (NSE-H)

Chen et al [15] proposed a framework to secure the VM migration. This framework is an extension on the Hypervisor. The whole system is called Network Security Engine-Hypervisor (NSE-H). This system has firewall, IDS function, and IPS function. Hence, the system will offer security to a virtual network. The architecture consists of Virtual machine migration agent (VMMA), Security Context Migration Agent (SCMA), Live Migration Coordinator (LMC), Network Security Engine (NSE), and hypervisor core.

D. Secured Live Migration - Role Based

Wang et al [16] introduced a framework based on policy-control in order to secure live migration. The framework depends on two key technologies (Intel vPro and TPM). This framework allows only authorized users to perform VM migration. This can be done in three main steps, which are building trustworthy containers for the VM, securing VM migration, and securing hypervisor core.

E. Improved Virtual Trusted Platform Modules (vTPM)

The concept of Trusted Platform Module (TPM) helps to secure sensitive information storage and permit verification of system integrity. Wan et al [17] proposed an improved secure vTPM migration protocol. The proposed protocol contains four phases: mutually authenticate, remote attestation of the destination, define security parameters by both source and destination, and create a secure session for transferring data.

F. Live Migration Defence framework (LMDF)

As discussed before, cloud providers have different data centers in different locations or even countries. This means they might migrate users' VM from one location to another without any notice from the user's perspective. Bedermann et al [6] developed a framework called LMDF, which will detect live migration from users' VM. This detection will help the user to delay the live migration in order to implement more security measurements before the migration of internal data. When the live migration is completed, the LMDF will be able to detect whether the VM was migrated to the same data center or to a new data center.

G. Inter-cloud VM Mobility

Nagin et al [18] introduced a new technology that involves inter-cloud proxies to secure channels between proxies, migration with non-shared storage, and virtual network migration. Inter-cloud proxies enforce hosts that are involved in inter-cloud VM mobility to keep their IP address private. Also, unauthorized users will be restricted from accessing hosts that are used by the inter-cloud VM mobility. Secure channels between proxies will provide an SSH tunnel between proxies in order to secure the VM migration. They identified three channels: secure inter-cloud migration channel, secure inter-cloud network, and secure inter-cloud storage channel.

H. Trust Cloud Security Level (TCSL)

Chen et al [19] proposed a new architecture for the cloud environment. The cloud environment will be divided into different zones and security levels. Each trusted zone and cloud has its own security level. The reliable migration module (RMM) will be responsible for managing the migration of the VM. This module has four main functions. First, central security management, which will manage security management of the resource and scheduling for all security levels. Second, security attributes, which are attached to the VMM and provide all security attributes, such as encryption and integrity check. Third, component security management, which will provide node controllers to separate nodes in different zones from each other. Forth, migration waiting queue is going to arrange the VMs' migration requests.

I. Secure VM migration by using RSA with SSL protocol

RSA is a public-key cryptosystem for securing data transmission. Varsha and G. Patil [20] proposed a methodology to secure live migration by using an RSA encryption mechanism with SSL protocol. The focus of their methodology is to secure the VM live migration from the

sender host to the destination host. They consider several stages to migrate a VM using RSA with SSL protocol. First, calculating the load of the physical host. Second, using per-copy or post-copy to migrate the memory from one host to another, as seen before. Third, use RSA for encryption contents of the VM (e.g., memory contents) and authentication.

J. Trust Token based VM migration protocol

Aslam et al [21] proposed a mechanism to secure VM migration. They implement this mechanism by using TPM and proposing a platform trust credential (Trust_Token), which will be implemented in every cloud to identify the Trust Assurance Level (TAL) of that cloud. The scheme of their proposal has several steps. First, set policy, where users of the cloud specify a migration policy for the management of VM migration. Second is implementing migration policy. When the users need to be migrated from one host to another trusted host, the policy is implemented by the cloud service provider. Third is auditing migration by cloud users. The cloud users will check whether their defined policy has been followed by the cloud server provider or not. If the TAL in the Trust_Token of the receiver host matches the TAL of user migration policy, then the migration will occur.

VII. Conclusion and Future Work

Live migration is a new technology that has many advantages. However, its security issues discourage many sectors from taking advantage of it. This paper has discussed the virtualization and VM live migration procedures. Moreover, possible threats in VM live migration have been examined. Then, security requirements have been discussed that must be implemented in different classes of migration in order to get secure live migration. Finally, some existing solutions to secure VM live migration have been discovered. In future work, a framework will be developed that will meet security principles (confidentiality integrity, availability, and authentication).

References

[1] K. Hwang, G. Fox, and J. Dongarra,"System modeling, clustering, and virtualization," in *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things.* Waltham, MA: MK, 2012. Ch.1, sec.1.3.4, pp. 34–193.

[2] N. Ahmad, A. Kanwal, and M. A. Shibli, "Survey on secure live virtual machine (VM) migration in Cloud," *Information Assurance (NCIA), 2013 2nd National Conference on*, Rawalpindi, 2013, pp. 101-106. doi: 10.1109/NCIA.2013.6725332.

[3] M. R. Anala, J. Shetty, and G. Shobha, "A framework for secure live migration of virtual machines," *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Mysore, 2013, pp. 243-248. doi: 10.1109/ICACCI.2013.6637178

[4] C. Clark et al. "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation* (NSDI'05), vol. 2, pp.273-286, 2005.

[5] R. Ahmad, A. Gani, S. Hamid, M. Shiraz, F. Xia, and S. Madani. "Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues". *The Journal of Supercomputing*, vol. 71., no.7, pp. 2473-2515, July 2015.

[6] S. Biedermann, M. Zittel, and S. Katzenbeisser, "Improving security of virtual machines during live migrations," *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, Tarragona, 2013, pp. 352-357. doi: 10.1109/PST.2013.659608

[7] D. Kapil, E. S. Pilli, and R. C. Joshi, "Live virtual machine migration techniques: Survey and research challenges," *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, Ghaziabad, 2013, pp. 963-969. doi: 10.1109/IAdCC.2013.6514357

[8] Xen, "Xen Hypervisor.", (2015), [online]. Available: http://www.xen.org/products/xenhyp.html [Nov 17, 2015].

[9] VMWare, "vSphere ESX and ESXi Info Center.", (2015), [online]. Available: vmware.com/products/vsphere/esxi-and-esx [Nov 17, 2015].

[10] M. R. Desai and H. B. Patel, "Efficient Virtual Machine Migration in Cloud Computing," *Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on*, Gwalior, 2015, pp. 1015-1019. doi: 10.1109/CSNT.2015.263

[11] J. Oberheide, E. Cooke, and F. Jahanian. "Exploiting Live Virtual Machine Migration," in *BlackHat DC Briefings*, Washington DC, February 2008.

[12] Y. Chen, Q. Shen, P. Sun, Y. Li, Z. Chen, and S. Qing, "Reliable Migration Module in Trusted Cloud Based on Security Level - Design and Implementation," *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, Shanghai, 2012, pp. 2230-2236. doi: 10.1109/IPDPSW.2012.275.

[13] M. Aiash, G. Mapp, and O. Gemikonakli, "Secure Live Virtual Machines Migration: Issues and Solutions," *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, Victoria, BC, 2014, pp. 160-165. doi: 10.1109/WAINA.2014.35.

[14] J. Shetty, A. M R and S. G, "A Survey on Techniques of Secure Live Migration of Virtual Machine". *International Journal of Computer Applications,* vol. 39, no. 12, pp.34-39, February 2012.

[15] C. Xianqin, G. Xiaopeng, W. Han, W. Sumei, and L. Xiang, " Application-Transparent Live Migration for Virtual Machine on Network Security Enhanced Hypervisor". *China Communications*, vol. 8, no. 3, pp. 32-42, 2011.

[16] W. Wang, Ya Zhang, B. Lin, X. Wu, and K. Miao, "Secured and reliable VM migration in personal cloud," *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, Chengdu, 2010, pp. V1-705-V1-709. doi: 10.1109/ICCET.2010.548537

[17] X. Wan, X. Zhang, L. Chen, and J. Zhu, "An improved vTPM migration protocol based trusted channel," *Systems and Informatics (ICSAI), 2012 International Conference on*, Yantai, 2012, pp. 870-875. doi: 10.1109/ICSAI.2012.6223146.

[18] K. Nagin et al. " Inter-cloud mobility of virtual machines," in *Proceedings of the 4th Annual International Conference on Systems and Storage (SYSTOR '11).* 2011, pp.1-12. DOI=http://dx.doi.org/10.1145/1987816.1987820

[19] Y. Chen, Q. Shen, P. Sun, Y. Li, Z. Chen, and S. Qing, "Reliable Migration Module in Trusted Cloud Based on Security Level - Design and Implementation," *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, Shanghai, 2012, pp. 2230-2236. doi: 10.1109/IPDPSW.2012.275

[20] V. Patil and G. Patil, "Migrating Process and Virtual Machine in the Cloud: Load Balancing and Security Perspectives ". *International Journal of Advanced Computer Science and Information Technology, vol. 1, no. 1, pp. 11-19, 2012.*

[21] M. Aslam, C. Gehrmann, and M. Björkman, "Security and Trust Preserving VM Migrations in Public Clouds," *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, 2012, pp. 869-876. doi: 10.1109/TrustCom.2012.256

Using DroidDream Android Malware Behavior for Identification of Other Android Malware Families

YeKung Kim†, Kathy J. Liszka ‡, and Chien-Chung Chan ‡

‡ *The University of Akron*
Department of Computer Science
yk33@zips.uakron.edu, liszka@uakron.edu, chan@uakron.edu

Abstract

Android malware is a very real threat for mobile device users. We provide an analysis of the DroidDream family of malware in order to identify uncommon attributes and behaviors that can be used to detect similar behavior in other malware families. We use Naïve Baysian classification to analyze the DroidDreamLight family and a dataset of popular Google Play applications. We found similar behaviors, including getIMEI, isPackageInstalled, getRawResource, and getIMSI between the two malware families; similarities did not exist between DroidDream and benign applications from Google Play store.

Keywords: Android malware, DroidDream, DroidDreamLight, malware analysis, static analysis.

1. Introduction

Mobile malware, particularly on the open-source Android platform, is a modern and common place threat. While statistics are difficult to pin down, it's estimated that there are as many Android devices infected as there are PCs running Windows [1]. This paper provides an analysis of an Android malware family, with the hypothesis that identifying information can be used to detect similar behaviors between other malware families. We specifically target the DroidDream malware family and use the DroidDreamLight as our test case to see if we can identify malware from that family as malicious based on knowledge of DroidDream. Although they are similar in name, they are completely different in their malicious techniques and objectives.

DroidDream, first seen in 2011, is what we call a "family" of malware because there are multiple versions that have evolved and broadened in functionality from what appears to be a base version. It manages to gain root privileges on the device and gathers personal information from the device and sends it off to a remote server. DroidDreamLight is also a family of malware that executes without user intervention. It attempts to download and install new applications. Both are classified as Trojans.

The paper is organized as follows. Related research is covered in section two. In section three, we give a description of static analysis of DroidDream. Section four details the experimental results using Naïve Bayesian classification to identify a malware from the DroidDreamLight family based on DroidDream attributes identified in section three. Conclusions and future work are drawn in the final section.

ISBN: 1-60132-445-6, CSREA Press ©

2. Related Work

In general, malware is analyzed either statically or dynamically. In [2] network traffic originating from an application is collected and compared to DNS blacklists because it is a common way for malware to obtain the IP address of their command and control servers. In [3] they use dynamic analysis to collect system call and event information in unknown applications. These are used to create signatures for malicious applications. A method was proposed in [4] to block the installation of applications that are deemed to be dangerous based on permissions requested or intent filter combinations. Moa, [5] proposes a static analysis system called DroidMat that looks at attributes such as permissions, and API calls from manifest and dex files using smali code. DroidMat can discriminate between malware and benign applications using machine learning techniques. Other static analysis approaches are described in [6] and [7] that are rely on manually crafted detection patterns which are often not available for new malware instances.

Dynamic analysis for Android malware is performed by executing programs on a real or virtual Android machine. TaintDroid [8], DroidRanger [9] and DroidScope [10] are methods that can monitor the behavior of applications at run time. They are very good at identifying malicious activity but they are required to run on the mobile device at the same time and thus require a significant amount of system resources, making it not practical.

3. Static Analysis of Android Malware

DroidDream is the first known Android malware to successfully gain root privileges and access a user's identification information. Once compromised, the infected phone can download additional malicious programs as well as leave the phone open to other attacks. It works in two phases, much like botnets on Windows platforms behave. In the first phase, DroidDream roots the device by breaking out of Android's security container and then downloads and installs a second application. In the second phase, it runs the downloaded app as a system application, which prevents the user from seeing or uninstalling the app without special permissions. This is the basis for a Command and Control (C&C) system. See [11] for more details on how the malware works.

DroidDreamLight was discovered in the same timeframe, around 2011. Although similar in name, it actually behaves quiet differently. It gathers information about the mobile device and then uploads the data to one or more websites. Malicious behavior is invoked on receipt of an *android.intent.action.PHONE_STATE* intent, meaning that it is not dependent on manual launch of the installed app to trigger its behavior.

Static malware analysis is the process of studying and reverse engineering an executable without actually loading and running it on a device. Dynamic malware analysis is the process of observing and studying the runtime behavior. We only use static analysis in this research by examining the AndroidManifest.xml file. This is found in the APK file, which is the package file format used to distribute and install applications on the Android operating system. It contains the Android manifest file, the program's code in classes.dex files, and the application's resources.

Androguard[1] is a reverse engineering tool for Android applications. Written in Python, it can be used to extract information from the APK file. There are other reverse engineering tools available but we chose this one because it is fast and easy to write customized Python scripts for it. Using this tool, we converted the APK files to a readable format, extracted keywords, and chose a set among those as attributes for our experiment.

[1] http://code.google.com/p/androguard/

We worked with sixteen unique DroidDream samples obtained from the Android Malware Genome Project [12]. Our first step was to identify permissions, intent-actions, intent-categories, and function names in the APK files. From this, we extracted a set of attributes with which we try to detect other malware. Figure 1 shows the attributes we selected after examining the malware. We considered permissions, intents, services, and functions. The only intents with a significant number of occurrences were too common to consider related to DroidDream, for example Android,intent,action MAIN which all apps use an an entry point. Therefore, no intents were included in our final list of attributes. In the following subsections, we discuss the other three areas we considered and the DroidDream behaviors we chose as attributes.

3.1 Permissions

Each Android application includes a manifest file that lists the permission requested by the application. When an app is installed, these permissions are displayed to the user who then decides whether to proceed with the installation or to cancel it. We looked at the permission classes present in the DroidDream dataset and determined the frequency that each appeared. Permissions that are known to be fairly common and unrevealing or that did not occur at least 10 times were excluded. We were looking for permissions that appeared in a super majority of the DroidDream APK files.

The following permissions were selected because they are associated with the known behavior of the malware, i.e., rooting the device and sending information to an external server.

- ACCESS_WIFI_STATE – Allows applications to access information about the WiFi network.
- CHANGE_WIFI_STATE – Allows applications to change WiFi connectivity state.
- INTERNET – Allows applications to open network sockets.
- READ_PHONE_STATE – Allows read only access to the phone state.

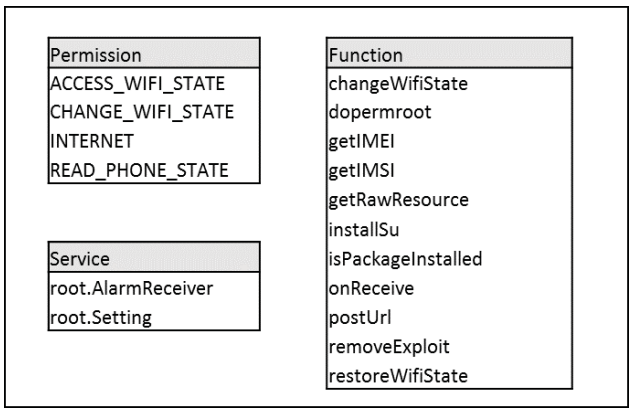


Figure 1. Selected DroidDream attributes

Other permissions considered were READ_CONTACTS and WRITE_CONTACTS which allow an application read and write access to the mobile phone's contact list. READ_LOGS allows an application to read the low-level system log files. ACCESS_NETWORK_STATE allows applications to access information about networks. While these are known behaviors of DroidDream, they did not occur frequently enough in the dataset to include them.

3.2 Services

A service is an application component that can perform long-running applications in the background. It does not provide a user interface and can continue to run in the background even when the user switches to another application. We found 15 instances each of

com.android.root.AlarmReceiver and *com.root.Setting*. These two services decrypt a byte buffer containing an IP address and URL of the service which is used to post data about the infected device [14].

3.3 Function names

We eliminated the functions *onlick*, *onCreate* and *onDestroy* as these are commonly used on Android apps, revealing no potential malicious activity. We also did not consider functions with obfuscated names: a, b, c, d, e, etc. There remained eleven functions that occurred in a supermajority of the dataset and had functionality that matched known DroidDream behavior. For example, the *getIMEI*, *getIMSI* and *getRawResource* functions collect user information. *Installsu* gets root permission on a device; *isPackageInstalled* checks whether an additional package is installed or not; *onReceive* collects additional information from the network; *postUrl* could be dangerous because it posts to a URL; *changeWiFiState* and *restoreWiFiState* are suspicious because they connect to WiFi without the user's knowledge. A function named *removeExploit* raises a red flag for obvious reasons.

4. Experiment and Results

In total, we identified 17 attributes we believe are strongly related to DroidDream. Next, we conducted a classification experiment to see if those would reveal any similarity to the other malware or simply benign applications. We tested with 46 DroidDreamLight samples collected from the same source [12] and 46 of the top free Android applications available on the Google Play[2] store.

We performed a Naïve Bayesian classification using Weka[3] software to evaluate the behavior of DroidDreamLight (DDL) and the top free Google applications (TOPS) based on the selected DroidDream attributes.

4.1 Naïve Bayesian Classification

The dataset was randomly split 66:34 into a training set and a test set. The results in Table 1 show that 90.3% of the instances are classified correctly. To classify each attribute value, a probability was assigned for each decision value. The Naïve Bayesian model shows that the relative frequency of class DLL is 50% (0.5) and the relative frequency of the TOPS class is also 50%. The frequencies for each class are multiplied with the attributes' probabilities for classification.

Table 1. Naïve Bayes Confusion Matrix

Actual Positive (DDL)	Actual Negative (TOPS)	
(True Positive) 12	(False Positive) 0	Predicted Positive (DroidDream)
(False Negative) 3	(True Negative) 16	Predicted Negative (Non-DroidDream)

In Table 2, we see each of the 17 DroidDream attributes and probabilities. For example, *restoreWifiState*, which NB did not select, doesn't appear in either the DDL dataset or the TOPS dataset. *CHANGE_WIFI_STATE* doesn't exist in 94% of DDL and 84% of the TOPS apps. *getIMEI* exists in 70% of the DDL dataset but only 4% of the TOPS group, and so forth.

From this result, we find an interesting fact that the shaded attributes such as *getIMEI*, *isPackageInstalled*, *getRawResource*, *getIMSI*, and *ACCESS_WIFI_STATE* act oppositely

[2] https://play.google.com/store

[3] http://www.cs.waikato.ac.nz/ml/weka/

between the DDL and TOPS apps. In other words, these attributes are mostly seen in the DroidDreamLight family but not in the top Google Play applications. However, *ACCESS_WIFI_STATE* is not seen much in DDL, but it is seen very frequently in TOPS.

To further analyze the output, we note that *getIMEI* and *getIMSI* are functions used to collect a phone's IMEI and IMSI information. This security sensitive information can be sent to a remote C&C server. The *getRawResource* function collects raw data, including product id, partner, model, SDK value, language, country, user id, and so forth. The *isPackageInstalled* function is useful for a malicious C&C server in order to manage to additional package installations on the infected device. These results indicate that the DroidDreamLight malware family's behavior is, indeed, similar to DroidDream's behavior.

The top Google Play apps show a higher probability of using *READ_PHONE_STATE* and *ACCESS_WIFI_STATE*. These two permissions are very common for both malware and benign Android applications.

5. Conclusions

In this research, we tested a method to detect similar Android malware through first, analyzing the APK files of the DroidDream malware family and then identifying probable malware attributes. Using Naïve Bayesian classification we identified similar, key behaviors in the DroidDreamLight family as opposed to the free Google Play store applications.

We began with the DroidDream family, but we plan to extend these datasets to a larger number of malware families. We are currently using J48 decision tree classification to find rules to classify Android malware. If we identify the specific behaviors of the malware families through static analysis of the APK files, we may use the information to detect new malware with similar attributes that has not yet been identified,. This could prove to be a very useful detection method in the case of polymorphic and metamorphic Android malware.

Table 2. Output of the NB Classification

Attributes		DDL	TOPS	DDL x 0.5	TOPS x 0.5
restoreWifiState	TRUE				
	FALSE	1	1	0.5	0.5
removeExploit	TRUE			0	0
	FALSE	1	1	0.5	0.5
CHANGE_WIFI_STATE	TRUE	0.06	0.16	0.03	0.08
	FALSE	0.94	0.84	0.47	0.42
postUrl	TRUE	0.02	0.04	0.01	0.02
	FALSE	0.98	0.96	0.49	0.48
getIMEI	TRUE	0.7	0.04	0.35	0.02
	FALSE	0.3	0.96	0.15	0.48
installSu	TRUE			0	0
	FALSE	1	1	0.5	0.5
dopermroot	TRUE			0	0
	FALSE	1	1	0.5	0.5
isPackageInstalled	TRUE	0.69	0.08	0.345	0.04
	FALSE	0.31	0.92	0.155	0.46
changeWifiState	TRUE			0	0
	FALSE	1	1	0.5	0.5
getRawResource	TRUE	0.69	0.02	0.345	0.01
	FALSE	0.31	0.98	0.155	0.49
getIMSI	TRUE	0.7	0.04	0.35	0.02
	FALSE	0.3	0.96	0.15	0.48
onReceive	TRUE	0.98	0.96	0.49	0.48
	FALSE	0.2	0.04	0.1	0.02
AlarmReceiver	TRUE	0.02	0.31	0.01	0.155
	FALSE	0.98	0.69	0.49	0.345
root.Setting	TRUE			0	0
	FALSE	1	1	0.5	0.5
INTERNET	TRUE	1	1	0.5	0.5
	FALSE			0	0
READ_PHONE_STATE	TRUE	0.98	0.69	0.49	0.345
	FALSE	0.02	0.31	0.01	0.155
ACCESS_WIFI_STATE	TRUE	0.08	0.73	0.04	0.365
	FALSE	0.92	0.27	0.46	0.135

6. References

[1] http://www.zdnet.com/article/mobile-malware-on-the-rise-worldwide-ransomware-hits-the-spotlight/.

[2] Iland, D. and Pucher, A., "Detecting Android malware on network level," in Proc. of MobiSys 2012, ACM, Lake District, UK, pp. 253–266.

[3] Isohara, T., Takemori, K., and Kubota, A., "Kernel-based behavior analysis for Android malware detection," in 2011 Seventh International Conference on Computational Intelligence and Security, pp. 1011 - 1015.

[4] Enck, W. and Ongtang, M., "On lightweight mobile phone application certification," in Proc. of ACM Conference on Computer and Communication Security, 2009, pp. 235-245.

[5] Wu., D., Mao, C., Wei, T., Lee, H., and Wu, K., "DroidMat: Android malware detection through manifest and API calls tracing," in 2012 Seventh Asia Joint Conference on Information Security, 9-10 August 2012, Tokyo, Japan, pp. 62 - 69.

[6] Felt, A. and Chin, E., "Android permissions demystified," in Proc. of ACM Conference on Computer and Communications Security, 2011, pp. 627-638.

[7] Grace, M. and Zhou, Y., "Riskranker:scalable and accurate zero-day android malware detection," in Proc. of International Conference on Mobile Systems, Applications, and Services, 2012, pp. 281-294.

[8] Enck, W. and Gilbert, P., "Taintdroid:An information flow tracking system for realtime privacy monitoring on smartphones," in Proc. of USENIX Symposium on Operating Systems Design and Implementation, 2010, pp. 393-407.

[9] Zhou, Y. and Wang, Z., "Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets," in Proc. of Network and Distributed System Security Symposium, 2012.

[10] Yan, L. and Yin, H., "DroidScope: seamlessly reconstructing the OS and Dalvik semantic analysis," in Proc. of the 21th USENIX Security Symposium, 2012, p. 29.

[11] Lookout, "Do Androids Dream….?" https://blog.lookout.com/blog/2011/03/06/do-androids-dream%E2%80%A6/, March 2011.

[12] Android Malware Genome Project. http://www.malgenomeproject.org/.

[13] Nakedsecurity, Aftermath of the Droid Dream Android Market malware attack, http://nakedsecurity.sophos.com/2011/03/03/droid-dream-android-market-malware-attack-aftermath/, March, 2011.

Anomaly Detection and Machine Learning Methods for Network Intrusion Detection: an Industrially Focused Literature Review

Colin Gilmore and Jason Haydaman
TRTech
100-135 Innovation Drive, Winnipeg, Canada.
colin.gilmore@trtech.ca, jason.haydaman@trtech.ca

***Abstract*—This paper outlines a literature review undertaken towards the goal of creating an industrial viable (real world) anomaly detection/machine learning based network intrusion detection system. We develop a taxonomy of available methods, and outline the pros and cons of each. This review leads to several important conclusions: (1) There are a large number of algorithms in the literature with significant level of overlap; (2) given the state of the literature today, it is not possible to objectively select the best algorithm; (3) there is a lack of research on the feature selection process needed for machine learning approaches; and (4) the low base-rate of attacks on computer networks compared with benign traffic means that effective detection systems will consist of many detection algorithms working simultaneously.**

Keywords— Network Intrusion Detection, Machine Learning, Anomaly Detection

I. Introduction

The amount of data stored on personal, industrial, and government computer networks is constantly growing. This creates a large incentive for other actors to attempt to illegitimately access these data. The economic cost of these attacks is notoriously difficult to quantify, however in 2011 the British Office of Cyber Security and Information Assurance estimated that cyber-crime cost the United Kingdom £27 billion per year, of which £21 billion was lost to espionage and intellectual property theft. The Canadian Cyber Security Strategy estimates identity theft losses (which, for the UK data represent only 6.3% of the total) costs Canadians $1.9 billion each year. In addition to the direct economic costs, an undetected network attack can have affects that go beyond economic, including the loss of confidence in a major government department or program.

The goal of our research team is to use anomaly detection and machine learning type approaches to improve the state-of-the-art for network intrusion detection. Our approach is an industrial one: we seek only to implement operationally viable algorithms (i.e. in operation in the real world). Thus, our approach to the problem is different than past reviews [1-8]. There has been significant effort in the academic literature relating to anomaly detection and data mining techniques for network intrusion detection (see [2,4,9]). However, this has not resulted in widespread industrial deployment. Two researchers note [9] "… despite extensive academic research one finds a striking gap in terms of actual deployments of such systems: compared with other intrusion detection approaches, machine learning is rarely employed in operational "real world" settings."

This work is the first part of taking up the challenge of creating a real-world deployment for an anomaly-detection/machine learning based network intrusion system.

The first step of such a process is to undertake a literature review of the available algorithms, and this paper outlines that process.

II. Intrusion Detection Overview:

Network intrusion detection can be divided into three types: signature based, specification based, and anomaly based [10].

1.1 Signature Based

Signature based techniques use a 'signature' – typically a hash – associated with a particular malicious activity. The most common signature based technique is an anti-virus program, which checks the signature of all files traversing a network, or being downloaded onto a computer. If the file being checked is a known virus/Trojan/worm, etc. then an alert is triggered. Signature based techniques have the advantage that there is very rarely a false alarm, but the disadvantage that they can (by definition) only detect known attacks. Significant effort must be made for signature management and subscription. Advanced attackers can readily avoid signature based detection because they are often capable of writing their own software, or may operate by taking over legitimate accounts.

1.2 Specification Based

Specification based techniques rely on the listing of network behaviors which are considered to be malicious. An example might be a brute-force login attempt. Snort is an example program that runs a specification-based detection engine [11]. Specification based techniques offer a more generalized way of detecting threats on a network, and may detect attacks not seen before, but often take a large amount of expert effort to specify. Unlike signature based detection, specification based detection can have false alarms. They can also be bypassed as these attackers will avoid obvious malicious behavior.

ISBN: 1-60132-445-6, CSREA Press ©

1.3 Anomaly/Machine Learning Based

Anomaly based techniques rely on detecting 'abnormal' or anomalous behavior. We include in this definition the various machine learning algorithms. These techniques take inputs from numerous network features, and label these features as 'anomalous' or 'normal' output. These techniques are the hardest for an attacker to avoid, as they are so general. However, they have the disadvantage of having high-false positive rates, which can make the detector useless in practical areas (discussed in Section III).

1.4 Notes on Features and Classification

In any detection system, there exist two main issues which need to be solved: feature selection and classification.

The features are the inputs which are selected as inputs to the algorithm. Features can include things like the Internet Protocol (IP) addresses, and much more. Classification is another word for 'which algorithm is used to determine which input data comes from a malicious source'.

1.5 Network Based Vs. Host Based Intrusion Detection

Network based anomaly detection algorithms depend only on data which is collected from network devices like firewalls, routers, Intrusion Prevention Systems (IPS), etc. Host based anomaly detection systems can include programs running on individual computers, which allows for more features to be added to the anomaly detection system. It is also possible to have a combined network and host-based system. Network based systems have the advantage of simplicity – there does not need to be a program running on every individuals computer (in some networks, it may be impossible to install a host-based agent on every computer). We consider only network-based systems in this review.

III. Base rates and the problem with False Positives

Axelsson [12] gives an excellent discussion about the problem with false positives for network intrusion detection. This issue is best illustrated with an example: Imagine that we have a network intrusion detection method where, if the detector sends an alarm, it has a 99% chance that it is a true network intrusion, and a 1% chance of a false alarm (positive). Next, assume that on a particular network, 1 in 10,000 input features comes from a malicious source. Now, assume that the anomaly detection system signals that it has detected an intrusion. What is the probability that the system has actually detected a real network intrusion?

The answer comes from Bayes theorem, and is quite low: 0,98% (or about 1%). Thus, when this system triggers, 99 times out of 100, it is a false alarm. This number is known as the positive predictive value of the test. Axelsson claims that the positive predictive value of the test must be above 50%, or most human operators will completely disregard the detector [12].

Due to this base rate problem, and the fact that the vast majority of network traffic is not malicious [12], care must be taken to keep false positives to a minimum. To obtain reasonable positive predictive values, the test must have a false positive rate on the order of the event we are trying to detect.

IV. Feature Selection

The selection of features is perhaps the most important part of any anomaly detection process. If the right features can be found, then there is no need for further detection processes. The best (ideal) feature for network intrusion detection would be a feature that is in one state when there was an attack and another state when no attack was occurring. If this ideal feature existed, there would be no 'anomaly' style classification algorithm necessary. We can also imagine the worst feature – in this case, the feature would not change at all when an attack occurs (it would have zero sensitivity to an attack). If the features we use as inputs to the anomaly detection system do not vary, then even the best algorithm will not be able to detect the intrusion. In practice, of course, the features extracted from network traffic will rest between these two extremes. The features used for network intrusion will vary somewhat for both normal and malicious traffic. The art of feature extraction to find features that vary a small amount for normal traffic, and then vary more significantly when an attack occurs. If there is a large enough difference in these features, then we can detect the attack with a suitable algorithm.

Ideally, we would like to select the minimal set of features that allow us to appropriately classify network behavior into normal and malicious categories. In practice, most features are selected on an ad-hoc basis based on expert domain knowledge. Automated feature selection procedures are available, but they will rely on a large data set of labelled training data [13] which is generally not available.

Based on [2], we have shown a taxonomy of features in Fig. 1. The features are split into network traffic based features, as well as network taxonomy features. Network taxonomy refers to the structure of a network (number of hosts, network organization). Network traffic can be split into three sub-groupings: flow data, protocol analysis, and derived features.

Flow data are data which capture which two computers are talking to each other at what time. A flow datagram will have source/destination IP and port numbers as well as protocol used for the communication.

Derived Features: We use the term derived features to mean features which are not readily available from the more basic data. A good example is Principal Component Analysis [14,15] – where a large number of (typically correlated) basic input variables are processed into a (much) smaller number of uncorrelated input variables. The outputs of this technique are variables which have no easy-to-grasp relationship to the more basic inputs. Other examples include syslog information, authorization logs, etc. [16].

Protocol analysis: many features we will use in this project fall under the protocol analysis label. Any feature which is not part of the flow data, or a 'derived' feature will be of the protocol analysis type. Examples include the browser agent used or the particulars of a DNS query.

Gonzalez [17] notes that "Identification of cyber attacks and network services is a robust field of study in the machine

learning community. Less effort has been focused on understanding the domain space of real network data in identifying important features for cyber attack and network service classification." Gonzalez presents a systematic way of making a set of derived features (13-27 input features) from a much larger set of basic features (over 200). The top features set include port numbers, packet length, number of truncated packets etc. Other research in this area includes [16,18].

V. Anomaly Detection and Machine Learning Methods

The use of anomaly detection algorithms for network intrusion detection has a long history. To the best of our knowledge, the use of anomaly detection for network intrusion detection began with Denning in 1987 [19].

There exists a large number of papers on anomaly detection: a thorough review of the experimental methods used between 2000-2008 found 276 peer-reviewed papers [20]. The huge number of papers in this research area means that we will almost certainly miss many papers. However, our approach has relied on both reading systematic reviews/taxonomies, and finding examples of each class of anomaly detection method.

We must be aware that many academic papers may have been published simply for the sake of being a novel approach. Given our goal of industrial deployment, we only concern ourselves with papers that offer industrially viable contributions.

In our literature search, we have come across 8 separate survey papers that relate to anomaly detection [1-8]. Most are focused on anomaly detection for network intrusion detection, and one more general, e.g. [1]. Much of this remaining section relies on the results of these surveys. We have summarized the methods discussed in this review (with their pros and cons) are outlined in Table 1.

A review of *evaluation* techniques used for proposed anomaly detectors [20] concludes that '[anomaly detection] studies from all categories fail to follow basic principles of scientific experimentation.' That is, the 'tests' used to prove the usefulness of a particular technique are not useful for actually evaluating the technique in an objective way. From our perspective this means it is impossible from to objectively decide which anomaly detection algorithm performs the best.

A. Anomaly Detection Categories

Garcia-Teodoro *et. al*. [4] split anomaly detection into several categories. We have created our own taxonomy, based on [2] and [4]. This is shown in Fig. 1. We note that these categories can have significant overlap. For example, some 'knowledge based' systems are also machine learning, and vice-versa. Further, almost all machine-learning and data mining approaches can be called statistical. Thus, this taxonomy is very loose at best.

Statistical Based: Network traffic is captured and a profile representing its stochastic behavior is created. Two datasets are considered: current and trained. Examples: Single and multiple variate, and histograms.

Knowledge Based: (e.g. an expert system). Classify the audit data according to a set of rules. Example: Finite state machines.

Machine Learning Based: Based on establishing an explicit or implicit model that enables the patterns analyzed to be categorized. Need for labelled data in all cases. Examples: Bayesian Network, Markov Models, Neural Networks, and Fuzzy Logic Techniques

Table 1: Pros and Cons of Anomaly Detection Methods

Method Name	PROS	CONS
Univariate /Multivariate	Simple to use, Can use on many parameters and correlations	High false positive rate; Unreasonable assumptions about underlying probability density functions; Likely to miss small changes in data
Histogram	Simple to use; Can deal with multiple time-scales quickly; Makes no assumptions about underlying statistics;	Due to averaging nature, likely to miss small changes in data; Need large amounts of data to work with
Finite State Machine	Contribution of human expert can limit number of states	Finite State Machines are a subset of Markov models; Need to define states, when states may be unclear
Expert System	Exact definition unclear: any type of human-influenced detector could be an 'expert system'	Vagueness of definition makes this hard to decide 'expert system' represents a method or not.
Association Rules/ Inductive Rules	Useful for identifying new patterns in data set; Unsupervised learning; Find patterns without human intervention	For rare events, high false positive rate; Need significant amount of data to reliably determine associations
Bayesian Networks	Good algorithms exist for training; Can encode causal relationships; Widely used Unsupervised learning	Results sometimes outperformed by much simpler methods; States must be defined by user; Similar to Markov Models
Markov Models	Can use Hidden Markov Models so states do not need to be known (only outputs); Unsupervised learning	Similar to other state-based approaches
Neural Networks	Unsupervised learning; No assumptions about statistical model	Supervised learning is desirable; Lots of data is required
Fuzzy Logic	Uses linguistic variables	Other ways of expressing uncertainty; Not widely used in literature
Genetic Algorithms	Capable of finding local minima; Possible to use to solve sub-problems of the overall anomaly detection algorithm	Very slow; Lots of computational resources; More efficient methods available
Clustering	Simple algorithms exist to make clusters; Unsupervised learning	False positives are likely for outlier events: as network traffic more diverse than typically thought
Support Vector Machines	Excellent method at finding separable classes; Low false positive rate; No assumptions about statistical model	Must have labelled training data; Difficulty in finding the function to map decision space onto new dimensions

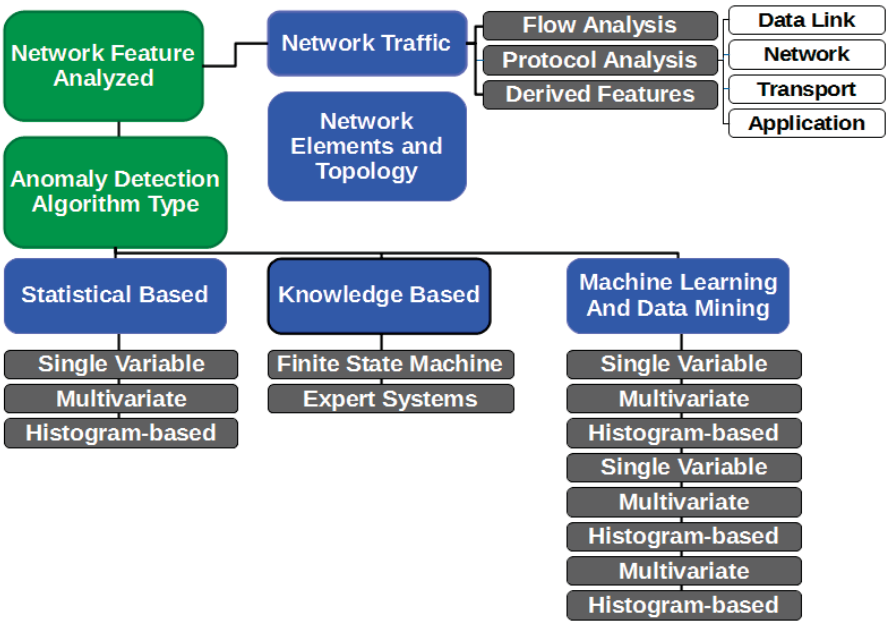


Figure 1: A Taxonomy of Anomaly Detection methods. Adapted from [2] and [4].

B. *Statistical Based Examples*

Statistical based models rely on creating some type of underlying model about a particular variable such as traffic volume, or number of connections per hour. One method would be to assume a particular probability density function for a variable, then calculate parameters such as mean and standard variation [19]. If these parameters are outside of some threshold, then an anomaly is triggered. Statistical techniques may also be practiced on more complicated derived features [21].

1) Single-variate, Multi-variate based

Single-variate and multi-variate models can be used for these statistical models. In the multi-variate case, correlations between multiple variables can be considered. [22, 23]

Despite all the research into mathematically more complicated detection methods, some claim that these simple techniques with thresholds often out-perform the more complicated methods. [2,24].

2) Histogram based

In most cases, one cannot assume an underlying statistical model for computer network traffic – e.g. variables do not fall into a simple Gaussian pattern. Considerer, e.g. a probability density function of the first 8 bits of an IP address (e.g. 196.xxx.yyy.zzz). [21] The probability density function will be dominated by the internal private IP addresses of the network, with large spikes located at '172', 192' and '10' (depending on how the internal network is configured).

In cases like these, a histogram-based approach is a viable solution. One can create a statistical model through histograms of the training data, and then compare the test data on the same histogram. Various metrics can then be used to generate 'normal' and 'anomalous' labels [21, 25, 26].

Histograms may also be constructed over various time-scales and for datasets with large differences in sizes. In this case, there is a significant problem of comparing histograms with different numbers of bins. This leads to a non-linear optimization problem [25,26].

C. *Knowledge Based Examples*

Knowledge based intrusion techniques rely on the use of a human expert to define a set of rules or process that are 'normal' or allowed. Deviations from these processes are flagged as anomalies. To a certain degree, our use of feature selection could be viewed as a knowledge- based approach to anomaly detection. We are encoding the expert knowledge in the feature selection process, rather than in the classification (anomaly detection) algorithm.

1) Finite State Machine / Markov Chain

Finite state machines (which also can be viewed as a Markov Chain) are an abstract system that transitions from one state to another with a certain probability [27,28]. Within the concept of a knowledge-based intrusion detection system, the model can be constructed by a human expert – who knows the states which should exist. The probabilities of transitions between states can then be determined from training data [19].

If the testing data exhibit low-probability state transitions, this can be flagged as an anomaly. In the context of machine learning, the same state-based model can be used, but the states, and the probability of transitions, are created via automated processes – not human experts.

2) Expert Systems

Expert systems are systems which attempt to emulate expert human reasoning via machines [29]. While a strict application of the term expert system involves the creation of a knowledge base and an inference engine [29], we feel that the term expert

system could be applied to almost any anomaly detection system, as the goal is to replace/assist a skilled human operator.

The Next Generation Intrusion Detection Expert System (NIDES) is an example of an expert system used for network intrusion detection [30], which has since evolved into a project called Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [31]. EMERALD offers a suite of anomaly detection tools and a signature engine.

D. Machine-Learning and Data Mining Examples

1) Association Rules/inductive rules

Association rules are a data-mining approach to anomaly detection. It takes a set of variables and discovers relations that commonly exist between the values. For example, it may be common in intrusion situations to have the events {virus detected, ftp attempt} -> {data loss}. Almost everyone has had experience with association rule algorithms, e.g. ('people who bought this book also bought ....') on websites like Amazon.

Examples of association rules for use in network intrusion detection include [32,33], where the authors use an association rule algorithm to detect abnormal record activity via unsupervised learning.

Association rules can work with both supervised and unsupervised data, and have the pros of being able to find their own rules. The cons of this type of method include the fact that rare events which cannot be trained for will trigger an 'anomalous' reading. This is likely to lead to a large number of false positives in a live computer network, which has much larger diversity than most people intuitively expect [9].

2) Bayesian Networks

A Bayesian network is a graphical model of a set of random variables, and the conditional dependencies of those variables on each other. Bayesian networks are directional – that is they imply causality by the direction of the relationships between variables. They are similar to Markov models, but Markov models do not have directional links between the states. Bayesian networks have been used for network intrusion detection. Examples include [34-36].

3) Markov Models

The Markov model-based intrusion detection systems try to calculate the likelihood of system in an anomalous state based on a sequence of observations. For example, a sequence of alerts from an Intrusion Detection System (IDS) such as Snort can be used to calculate the probability of system being under attack.

To understand this technique better, a brief description of Markov model is presented. (from [37]) A Markov model is a statistical model with N states S_1, S_2, ..., S_N and discrete timestamps. On a given timestamp, system is in exactly one of N states and between timestamps states are chosen randomly. An important property of a Markov model is that being at state S_t on timestamp t only depends on the system's state on timestamp t-1 and all the earlier states do not have any effect in selecting state S_t.

An extension to the Markov model is Hidden Markov Model (HMM) in which a system's states and state transitions are not visible and only events from these states are observable. Formally speaking, an HMM is a five-tuple (N,M,Π,A,B) where N is the number of states, M is the number of possible observations, Π is the starting state probabilities, A is the state transition probability matrix, and B is the observation probability matrix. Since HMM is a statistical model, the initial values of Π, A, and B are selected randomly and a training process is required to make this model ready for actual data.

For example in [38], a computer network can be in one of Normal, Attempt, Progress, or Compromise states. These states are not known at a given time and only Snort events are observed. Snort events can be categorized into different groups based on their severity to limit the number of observable events. Initially we can assume the system is in Normal state and initialize state transition matrix and observation probability matrix with random numbers. Once the HMM is trained, an intrusion detection system can find the probability of system being in one of states based on sequence of observed Snort events.

4) Neural Networks

In cases where a relationship between inputs and outputs is expected, but the exact relationship between the inputs and outputs is unknown, Artificial Neural Networks (ANN) (or just Neural Networks) are useful. ANN's are based on a simple model of how neurons work in the human brain. There are many nodes (or neurons) operating in parallel. Each node is connected to other neurons in the next layer by a specific weight. Changing the weights allow the network to model very complex, multidimensional functions – in practice this means that the differences between normal and anomalous behavior can be quite complex . In order to set the weights of each node, ANN's require large sets of supervised learning.

Neural networks have been used in various ways for network intrusion detection [39-42].

5) Fuzzy Logic

Some authors have used fuzzy logic approaches to network intrusion detection e.g. [43,44]. In fuzzy logic, other categories – rather than true or false – can be used for a set. The truth value of a fuzzy set lies somewhere between 0 and 1. Fuzzy logic has received some criticism as a tool [45], and some defense [46]. Pros of fuzzy logic include the ability to have non-conclusive outputs, but cons include high computational costs, and a lack of clarity on the proper approach with fuzzy logic (e.g. why not just use a percent output from the classification algorithm, with a probability from 0 to 1?).

6) Genetic Algorithms

Genetic algorithms are an optimization technique which is modelled on the process of biological evolution. Advantages of this optimization technique include the fact that it is capable of finding a global minimum in an optimization function with a large number of local minima, and the lack of assumptions required to use this algorithm. The largest disadvantage is that the huge amount of computational resources required to find the global minimum. These techniques have been used with respect to intrusion detection [47]. Genetic algorithms could be used to optimize the parameters for other detection algorithms [48,55]. In cases such as these, a genetic algorithms can be used to find the optimal parameters for the Support Vector Machine (e.g. the selection of the mapping function in the SVM).

7) Clustering (K-means)

The k-means algorithm [49] is a centroid-based partitioning technique that takes the input parameter *k* and partitions *n* objects into *k* clusters so that an object is 'similar' to other objects within a cluster and different than objects in the other clusters. This algorithm thus groups like objects together automatically, in an unsupervised learning environment. To find the similarity between objects, a set of features are selected and distance between these features are measured. Initially, the k-means algorithm randomly selects k objects representing k clusters. As an example, Munz et al. [50] used network flow as the source for anomaly detection and selected total number of packets sent from/to a given port, total number of bytes sent from/to a given port, and number of source-destination pairs matching the given port number as features to compute similarity between different flows.

8) Support Vector Machines

Support Vector Machines operate as a group classifier by constructing a hyperplane or set of hyperplanes in a high dimensional space using training data to separate data into two groups of normal and abnormal (or malicious) classes. They generally require labelled training data. There are many examples of SVM's used in network intrusion detection (e.g., [52,53,54]).

One of the main advantages of SVM is its ability to discriminate data sets which are not readily separable by simpler techniques. If the 'normal' and 'abnormal' data sets to discriminate are not linearly separable or variations of features for two classes have overlaps, it maps the original data space into much higher space to make the separation easier. The function for mapping of features and the other parameters would be optimized under optimization methods. In practice, this means that input features which first appear to be poor indicators of malicious or abnormal activity may be good indicators with the SVM.

Another advantage of Support Vector Machines is that other applications have seen a low false positive rate [51]. Perhaps the biggest cost with SVM's are their complexity – in order to perform the non-linear mapping of the input feature space into the higher dimensional space, a particular mapping function must be selected and this often requires optimization techniques. This leads to long training times for the SVM.

VI. Conclusions

Through this literature review, we have reached several conclusions about anomaly detection and machine learning algorithms for use in real-world network intrusion detection systems.

- While many different anomaly detection approaches are outlined in the literature, there is significant overlap between many of them. For example, an 'expert system' could describe virtually any algorithm that has its initial inputs generated by a human expert, and Finite State Machines, Bayesian Networks, and Markov Networks are all extremely similar (and in some cases are sub-classes of each other)
- Due to the false positive problem when detecting rare events, and the fact that anomaly detection systems commonly have high false-positive rates, it is likely that a functional detection system will be comprised of several correlated detectors. This could include correlations with signature and specification based intrusion detection techniques.
- Given the lack of literature on feature selection, ad-hoc, expert supervised feature selection based on previous records of attacks will be the best method to generate relevant features.
- We suspect that a significant number of publications suggest algorithms which are not industrially viable.
- The evaluation of intrusion detection techniques is a problem from a scientific perspective [2]. This makes it very difficult to objectively determine which algorithm is the 'best'.

Acknowledgements

The authors would like to thank the Canadian Safety and Security Program at Defence Research and Development Canada for project funding.

References

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM Computing Surveys (CSUR)* 41.3 (2009): 15.

[2] Estevez-Tapiador, Juan M., Pedro Garcia-Teodoro, and Jesus E. Diaz-Verdejo. "Anomaly detection methods in wired networks: a survey and taxonomy."*Computer Communications* 27.16 (2004): 1569-1584.

[3] Axelsson, Stefan. *Intrusion detection systems: A survey and taxonomy*. Vol. 99. Technical report, 2000.

[4] Garcia-Teodoro, Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." *computers & security* 28.1 (2009): 18-28.

[5] Lazarevic, Aleksandar, et al. "A comparative study of anomaly detection schemes in network intrusion detection." *Proc. SIAM* (2003)

[6] Wegner, Ryan. Multi-Agent Malicious Behaviour Detection,. PhD Thesis, Department of Computer Science, University of Manitoba, 2012.

[7] Patcha, Animesh, and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends." *Computer Networks* 51.12 (2007): 3448-3470.

[8] Tsai, Chih-Fong, et al. "Intrusion detection by machine learning: A review."*Expert Systems with Applications* 36.10 (2009): 11994-12000.

[9] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in 2010 IEEE Symposium on Security and Privacy, 2010.

[10] Liao, Hung-Jen, et al. "Intrusion detection system: A comprehensive review."*Journal of Network and Computer Applications* (2012).

[11] www.snort.org, accessed March 2016.

[12] Axelsson, Stefan. "The base-rate fallacy and the difficulty of intrusion detection." *ACM Transactions on Information and System Security (TISSEC)*3.3 (2000): 186-205.

[13] Dash, Manoranjan, and Huan Liu. "Feature selection for classification."*Intelligent data analysis* 1.3 (1997): 131-156.

[14] Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Mining anomalies using traffic feature distributions." *ACM SIGCOMM Computer Communication Review*. Vol. 35. No. 4. ACM, 2005.

[15] Wang, Wei, and Roberto Battiti. "Identifying intrusions in computer networks with principal component analysis." *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. IEEE, 2006.

[16] Chebrolu, Srilatha, Ajith Abraham, and Johnson P. Thomas. "Feature deduction and ensemble design of intrusion detection systems." *Computers & Security*24.4 (2005): 295-307.

[17] Jose Andres Gonzalez, Numerical Analysis for Relevant Features in Intrusion Detection. MSc. Thesis, Department of the Air Force, Air University, Air Force Institute of Technology, Write-Patterson Air Force Base, Ohio, 2009.

[18] Mukkamala, Srinivas, and Andrew H. Sung. "Feature selection for intrusion detection with neural networks and support vector machines." *Transportation Research Record: Journal of the Transportation Research Board* 1822.1 (2003): 33-39.

[19] Denning, Dorothy E. "An intrusion-detection model." Software Engineering, IEEE Transactions on 2 (1987): 222-232.

[20] Tavallaee, Mahbod, Natalia Stakhanova, and Ali Akbar Ghorbani. "Toward credible evaluation of anomaly-based intrusion-detection methods." *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 40.5 (2010): 516-524.

[21] Kind, Andreas, Marc Ph Stoecklin, and Xenofontas Dimitropoulos. "Histogram-based traffic anomaly detection." *Network and Service Management, IEEE Transactions on* 6.2 (2009): 110-121.

[22] Ye, Nong, et al. "Multivariate statistical analysis of audit trails for host-based intrusion detection." Computers, IEEE Transactions on 51.7 (2002): 810-820.,

[23] Ye, Nong, et al. "Probabilistic techniques for intrusion detection based on computer audit data." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 31.4 (2001): 266-274.

[24] Kruegel, Christopher, et al. "Bayesian event classification for intrusion detection." Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003

[25] Beigi, Mandis S., et al. "Anomaly detection in information streams without prior domain knowledge." *IBM Journal of Research and Development* 55.5 (2011): 11-1.

[26] Beigi, Mandis, et al. "Muti-scale temporal segmentation and outlier detection in sensor networks." Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on. IEEE, 2009.

[27] "Finite-state Machines." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Accessed March 2016., http://en.wikipedia.org/wiki/Finite-state_machine

[28] "Markov Chain." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Accessed March 2016., http://en.wikipedia.org/wiki/Markov_chain

[29] "Expert System." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Accessed March 2016., http://en.wikipedia.org/wiki/Expert_system

[30] Anderson, Debra, Thane Frivold, and Alfonso Valdes. Next-generation intrusion detection expert system (NIDES): A summary. SRI International, Computer Science Laboratory, 1995.

[31] http://www.csl.sri.com/projects/emerald/, accessed March 2016

[32] Das, Kaustav, Jeff Schneider, and Daniel B. Neill. "Anomaly pattern detection in categorical datasets." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[33] Das, Kaustav, and Jeff Schneider. "Detecting anomalous records in categorical datasets." Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007.

[34] Bronstein, Alexandre, et al. "Self-aware services: Using bayesian networks for detecting anomalies in internet-based services." Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on. IEEE, 2001

[35] Kruegel, Christopher, et al. "Bayesian event classification for intrusion detection." Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003.

[36] Barbara, Daniel, Ningning Wu, and Sushil Jajodia. "Detecting novel network intrusions using bayes estimators." First SIAM Conference on Data Mining. 2001.

[37] Moor, Andrew. "Statistical Data Mining Tutorials", http://www.autonlab.org/tutorials/, accessed March 2016.

[38] Shameli Sendi, Alireza, Michel Dagenais, Masoume Jabbarifar, and Mario Couture. "Real Time Intrusion Prediction based on Optimized Alerts with Hidden Markov Model." Journal of Networks 7, no. 2 (2012): 311-321.

[39] Ryan, Jake, Meng-Jang Lin, and Risto Miikkulainen. "Intrusion detection with neural networks." Advances in neural information processing systems. MORGAN KAUFMANN PUBLISHERS, 1998.,

[40] Mukkamala, Srinivas, Guadalupe Janoski, and Andrew Sung. "Intrusion detection using neural networks and support vector machines." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 2. IEEE, 2002.,

[41] Zhang, Zheng, et al. "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification." Proc. IEEE Workshop on Information Assurance and Security. 2001.

[42] Cannady, James. "Artificial neural networks for misuse detection." National information systems security conference. 1998.

[43] Dickerson, John E., and Julie A. Dickerson. "Fuzzy network profiling for intrusion detection." Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American. IEEE, 2000.

[44] Gomez, Jonatan, and Dipankar Dasgupta. "Evolving fuzzy classifiers for intrusion detection." Proceedings of the 2002 IEEE Workshop on Information Assurance. Vol. 6. No. 3. New York: IEEE Computer Press, 2002.

[45] Haack, Susan. "Do we need "fuzzy logic"?." International Journal of Man-Machine Studies 11.4 (1979): 437-445.

[46] Zadeh, Lotfi A. "Is there a need for fuzzy logic?." Information Sciences 178.13 (2008): 2751-2779

[47] Li, Wei. "Using genetic algorithm for network intrusion detection." Proceedings of the United States Department of Energy Cyber Security Group (2004): 1-8.

[48] Bridges, Susan M., and Rayford B. Vaughn. "Fuzzy data mining and genetic algorithms applied to intrusion detection." Proceedings twenty third National Information Security Conference. 2000.

[49] Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C 28 (1): 100–108. JSTOR 2346830.

[50] Münz, Gerhard, Sa Li, and Georg Carle. "Traffic anomaly detection using k-means clustering." Proc. of Leistungs-, Zuverlässigkeits-und Verlässlichkeitsbewertung von Kommunikationsnetzen und Verteilten Systemen 4 (2007).

[51] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: A Multiple Classifier System for Accurate Payload based Anomaly Detection," Computer Networks, vol. 53, no. 6, pp. 864—881, Apr. 2009

[52] Mukkamala, Srinivas, Guadalupe Janoski, and Andrew Sung. "Intrusion detection using neural networks and support vector machines." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 2. IEEE, 2002.,

[53] Horng, Shi-Jinn, et al. "A novel intrusion detection system based on hierarchical clustering and support vector machines." Expert systems with Applications 38.1 (2011): 306-313.

[54] Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications, 39(1), 424-430.

[55] Kim, Dong Seong, Ha-Nam Nguyen, and Jong Sou Park. "Genetic algorithm to improve SVM based network intrusion detection system." Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on. Vol. 2. IEEE, 2005.

Using Football Formations in a Honeypot Environment

Sebastian Kollmannsperger and Tyrone S. Toland
Department of Informatics
University of South Carolina Upstate
800 University Way, Spartanburg, SC 29303
Kollmans@email.uscupstate.edu, ttoland@uscupstate.edu

Abstract — Unauthorized access to information continues to be a challenging problem, especially in a time where cyber attacks are on the rise. Current security measures (e.g., access control systems, firewalls, intrusion detection systems) may not be sufficient to protect the information technology (IT) infrastructure from a resourceful malicious attacker. This paper presents a novel approach to embed a football formation into a Honeypot environment. We show how executing football plays in a Honeypot environment can be used to gather information about a malicious attacker. This reconnaissance information can be used to prevent future unauthorized access to sensitive information. We also discuss of our implementation and provide some results from a proof of concept experiment.

Keywords — Honeypots, Intrusion Detection System, Information Security

I. INTRODUCTION

Information security has been challenging since humans began exchanging information. For example, cipher has always been discussed in information security. In fact, ciphers were used to encrypt important messages as far back as 50 BC [5]. The advent of the computer required stronger measures to enforce security, which became an even bigger challenge with the rise of the Internet. As companies become inter-connected more and more via the Internet, the challenge of protecting the infrastructure and information becomes an even bigger challenge. Nowadays many different defense mechanisms work together to form a secure system. Firewalls, encryption tools, access control systems, intrusion detection systems as well as other security software contribute to information security in a slightly different way.

Schneier [3] identifies three tasks of information security which are prevention, detection and response. All security tools can be assigned to either one of these tasks.

Prevention is the attempt to protect resources from danger and harm. Preparations have to be done, to set up mechanisms that protect the IT. The goal is to make it as hard as possible, for intruders and hackers to access resources. Well known prevention tools are firewalls, password protections, encryption tools and digital signatures.

When prevention is not effective, detection becomes an important process. With detection, we want to find out if our system was compromised and from where. Detection is therefore like a monitoring tool. However, it does not contribute to the protection of systems, because detection tools act rather passive. An intrusion detection system is an example for a detection system.

After an intruder has been detected, we have to react. Every action in a system gets recorded and stored by one of the detection tools. Therefore, also the intruder leaves behind evidences. By analyzing these evidences, we can find out how the attacker got in, what the attacker accessed and what the intruder manipulated. With this information we can take steps to react adequately. Backup and recovery tools are an example of response tools.

We now discuss a tool that can be used to assist in securing a computer system.

A. Honeypots

Compared to other approaches to information security, honeypots are a more aggressive and active form of defense against malicious attacks [2]. Honeypots are defined in different ways. Schneier [3] defines a Honeypot as a security resource whose value lies in being probed, attacked or compromised. This paper defines a Honeypot as an IT resource with the goal to attract potential malicious attackers. That is, any access of the Honeypots is examined and recorded to be used to deter similar attacks from occurring in the future. Contrary to other components of an IT system, it is desired that the Honeypot gets attacked and probed. Since Honeypots are masquerading as sensitive resource, they do not provide any functionality for an organization. Therefore, if a malicious user accesses the Honeypot, then this access can be seen as unauthorized access and therefore as an intrusion [2]. Honeypots can be categorized as either a production honeypot or a research honeypot as follows [3][4]:

- *Production Honeypot:* According to the name, these kind of Honeypots are especially used in a production environment. Their main purpose is to gather information for a specific organization about intrusions. They add value to an organizations information security.

- *Research Honeypot:* These Honeypots are used principally in a research environment to gather information about potential attackers. They do not add value to a specific organization. Information from Research Honeypots can be used to find out about techniques and resources from attackers

which can help to prepare the production system for attacks.

B. Value of Honeypots

Honeypots are flexible tools and contribute to each one of the three security aspect as follows [4][3]:

- *Prevention:* Contrary to the belief of the majority, Honeypots can help to prevent attacks because of deception and deterrence. Deception means, that potential attackers may waste time and resources on honeypots. Without knowing, attackers interact with a honeypot that imitates a valuable resource. During this interaction, organizations have the time to react. After all, attacks can be stopped before even leaking information. Deterrence on the other hand is the effect of scaring off attackers because of the warning effect of Honeypots. When attackers know that an organization uses Honeypots, they may not even try to attack. As we can see, honeypots contribute to the prevention of attacks in a certain degree. Nonetheless, traditional prevention tools like firewalls are more efficient.

- *Detection*: Honeypots have the biggest impact in detection. For many organizations, detection is a difficult topic. Schneier [3] identifies three challenges when it comes to detection: false positives, false negatives and data aggregation. False positives are mistakenly reported alerts. This happens, when the system interprets normal network traffic as an attack. The opposite false negatives are attacks, that the system does not notice. Finally, data aggregation is the struggle to collect the data and transform it into valuable information. Common intrusion detection systems struggle in these three aspects. Intrusion detection systems act like a watchdog over a company's IT infrastructure. They monitor the traffic and identify whether an access is authorized or not. Therefore, intrusion detection systems generate a lot of data, resulting in an overload of information. Honeypots however, help us to eliminate these negative aspects. Because every interaction with a honeypot can be seen as unauthorized, honeypots only register these interactions. The problem with data aggregation and false positives can be eliminated. False negatives can still occur, for example if an intrusion does not affect the Honeypot, but this risk can be mitigated by placing the Honeypot in an attracting position. Consequently, Honeypots help us to detect intrusions more effectively.

- *Response:* After an intrusion is detected, response is the next step to take. Honeypots help us to identify evidences via log files. That is, the user can analyze log files that are generated by Honeypots to find out how the attacker gain access to the system. With the information collected by a Honeypot, we can construct countermeasures to prevent similar attacks from occurring in the future.

It should be noted, that the goal of a Honeypot is not to prevent attacks, but to detect them. Therefore, a Honeypot should be combined with other security tools (e.g., firewalls, encryption, password protection).

In this paper we discuss how American football plays can be used to gather information about malicious attackers in a honeypot research environment. In particular, we propose using various offensive plays to provide valuable reconnaissance information to defend sensitive information in an infrastructure. This reconnaissance information can be analyzed and used to defend sensitive information in an infrastructure. Our novel approach to mapping football formations into a honeypot research environment can be extended to a networked infrastructure.

This paper is organized as follows. In Section II, we discuss a research Honeypot environment. In Section III, we briefly describe a simplified football formation. In Section IV, we show how to map a football formation into a research Honeypot environment. Section V discusses our implementation and results from proof concept experiment. Section VI concludes the paper.

II. RESEARCH HONEYPOT ENVIRONMENT

Honeypots allow a wide range of application areas. Because of their goal to distract and attract attackers, the best way to use Honeypots is within an IT infrastructure. The probability that an attacker interacts with Honeypots are increased by masquerading as sensitive data.

In Fig. 1, we illustrate a Honeypot integrated with other important and possible sensitive IT resources. Fig. 1 is a variation of a model in [4].

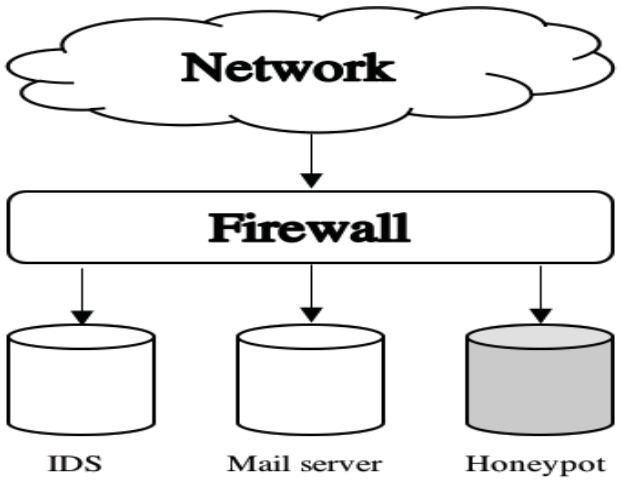


Fig. 1 Honeypot in IT Infrastructure

The Honeypot is part of the infrastructure similar to other important resources (e.g., mail server, web server). Therefore, the Honeypot distracts and attracts malicious attackers. Assuming that the attacker scans either our web or mail server, the likelihood that the attacker will access the Honeypot is high.

III. FOOTBALL OVERVIEW

We now provide a brief overview of American Football (football). In American football there are two teams of eleven players. Each team takes turns defending their goal. That is, the defending team wants to prevent the opposing team from taking the football into their end zone to score (e.g., touchdown, field goal, touch back).

A. Football Offensive Formation

Although in real football there are eleven players per team, we will only consider seven players in this paper. Our offensive formation consists of five players that form the offensive line (OL + ROL). The offensive line has the task of keeping the ball away from the defending team. Behind the offensive line we have the Quarterback (QB) and Running Backs (RB). The job of the QB is to control the play. The Running Back on the other hand tries to outrun the defense. Fig. 2 shows the offense represented a circles.

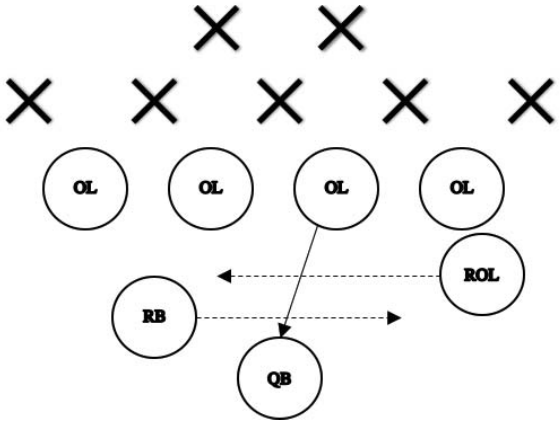


Fig. 2 Offensive & Defensive Formation

B. Football Defensive Formation

The defensive formation consists of five defensive linemen and two Linebacker (LB). The defensive linemen try to attack either the QB or the ball carrier. The LB are there to provide additional support for the defense. Sometimes the LB also try to sack the opposing QB. Ultimately, the goal of the defense is to get the ball and stop the attack. Fig. 2 shows the defense represented as X.

C. Double Reverse Flea Flicker

The double reverse flea flicker is one of many different football plays. It involves three players, the QB, the RB and one player of the OL, called the right offensive lineman (ROL). For the purpose of this play, the ROL positions on a different position to be able to perform the play. Fig. 2 shows the starting position. The dashed lines show the running paths of the players. The continuous line shows the path of the ball. So in the first move, the ball travels from the center of the offensive line to the QB. The ROL and RB run their paths. In Fig. 3 we can see the subsequent moves. When the RB crosses the QB, the ball travels from the QB to the RB (1). The next move happens, when the RB crosses the ROL. The ball travels from the RB to the ROL (2). The final move happens, when the ROL crosses the QB. Hereby, the ball goes from the ROL to the QB (3). During the play, the QB does not switch the place. However, the RB and the ROL cross and switch their sides. The ball travels from the center to the QB to the RB to the ROL and back to the QB. The goal of that play is to distract the defenders and create room for the QB to pass the ball. The opponent cannot recognize where the ball is and tackle the wrong player. This distraction has a huge similarity with the way Honeypots work. This is the reason why we chose to map this play onto a Honeypot research environment.

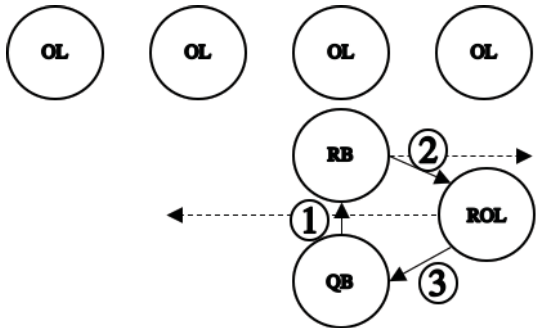


Fig. 3 Double Reverse Flea Flicker Moves

IV. COMBINING FOOTBALL FORMATIONS and HONEYPOTS

Fig. 4 shows how a football formation can be implemented into a research Honeypot environment. As explained in Section 3, Defenders are represented as X's and Attackers are represented as O's. We now map the football formation onto an IT infrastructure whereby the roles change. Now, the Attackers are X's (i.e., they retrieve something) and the Defenders are O's (they protect something). Therefore, in this model the football attackers are playing the role of the defense, while the football defenders play the role of the attackers.

The goal now is to protect the ball instead of carrying the ball into the end zone. In our model the ball represents the sensitive data. The Honeypots are masquerading the sensitive data to attract attackers. The defense are protection tools like Firewalls, encryption tools and password protection. The defense protects our infrastructure. This infrastructure consists of three Honeypots (HP1, HP2, and

HP3). Since we are working with a research environment, we do not have any production entity. The arrows illustrate unauthorized access. Since every defense mechanism is not completely safe, there may be some traffic coming through the firewall that will access the Honeypots. When this happens, the Honeypots will work together to execute the play in Section 2.3.

A. Running the Play in a Honeypot Environment

In Fig. 4, HP1 acts like the RB, HP2 acts like the QB and HP3 acts like the ROL. This means, that in the beginning HP2 (i.e., QB) masquerades as a sensitive resource (i.e., ball). So, the attackers try to access HP2. When this happens, we want HP1 to masquerade as a sensitive resource. So, we pass the ball to HP1. This again means that attackers now try to access HP1. Then, we want HP3 to masquerade as a sensitive resource, meaning HP3 becomes the new goal for attackers. Finally, HP2 again masquerades as a sensitive resource. To pass the data between the Honeypots, we will simulate data being active and inactive, which in essence we are not really passing data between the Honeypots.

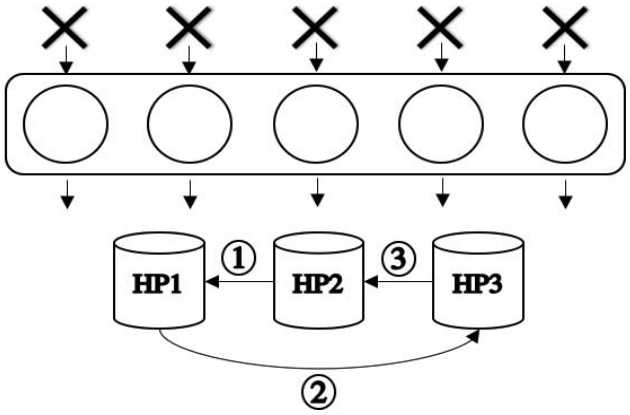


Fig. 4 Football Formation mapped in Honeypot Environment

V. IMPLEMENTATION and EXPERIMENT

We ran an experiment using a framework we implemented in Java 8.

A. Implementation

To show a proof of concept, we developed the following three programs:

- HoneypotServer (HPTS) – is a program that simulates the honeypot. The program uses a Boolean variable (e.g., activeData) to simulate access to the sensitive data (i.e., the honey). If activeData is true, then the access to sensitive data is available via HPTS; otherwise, if activeData is false, then the sensitive data is currently not available via access of this machine. That is, the sensitive data access has moved to a different machine.

- HoneypotManager (HPTM) – is a program that sends a message to either activate or deactivate access to sensitive data on HPTS. When data access has been deactivated on a HPTS, then the data access is activated on another HPTS, i.e., data access has moved.

- HoneypotAttacker (HPTA) – is a program that the attacker uses to attempt to access sensitive data on a HPTS. The attacker sends an access message request (i.e., a malicious attack message) to the HPTS. If HPTS has access capability to the sensitive data (i.e., activeData is true), then an active message is generated that contains: A (i.e., access to sensitive data is active), HPTA IP address, the attack message arrival time on HPTS, and the attack message departure time from HPTS. Otherwise, an inactive message is generated that contains: N (i.e., access to sensitive data is not active), HPTA IP address, the attack message arrival time on HPTS, and the attack message departure time from HPTS.

B. Experiment

We ran our experiment in a test networking lab. To simulate the example in Section IV, we ran HPTS on three computers (i.e., HP1, HP2, and HP3). We ran HPTA on a separate computer to simulate the attacks. On another separate computer, we ran HPTM to activate and deactivate data access on HP1, HP2 and HP3, respectively. For our experiments, HPTS only listens on port 9001.

This experiment implements Fig. 4. That is, HP2 is initially activated, while HP1 and HP3 are deactivated. The attacker can now search for the active honeypot using HPTA. To accomplish this, the attacker successively tries to connect to the honeypots. Once the attacker finds the active honeypot (i.e., activeData is true), the manager deactivates that honeypot (i.e., activeData is set to false) and then activates the next honeypot in the sequence. Then, the attacker searches for the next active honeypot and the process continues per Fig. 4.

Table 1 shows the result from this experiment. The attacker does follow the sequence of the play in Fig. 4 when accessing active data items. As we proposed, we could gather information from the malicious user in Msg 2 at HP2, in Msg 4 at HP1, in Msg 7 at HP3 and in Msg 8 again at HP2. That is, we can gather reconnaissance information from a malicious user at a given machine at a specified time.

C. Discussion

The experiment shows that our approach is feasible. Our approach provides a guaranteed time interval for which we can evaluate malicious activity. In particular, we can evaluate malicious activity when accessing an active

honeypot and/or when searching for an active honeypot. Based on Table I, we have extracted a set of active Honeypot access times (1) and a set of time intervals to search for an active Honeypot (2).

TABLE I EXPERIMENTAL RESULTS WITH TIMES IN MILLISECONDS

Msg#	HP#	Active	IP Address	ArrivalTime	DepartureTime
1	1	N	192.168.1.100	1463775948262	1463775948262
2	2	A	192.168.1.100	1463775950737	1463775950737
3	3	N	192.168.1.100	1463775957258	1463775957258
4	1	A	192.168.1.100	1463775958262	1463775958262
5	2	N	192.168.1.100	1463775966977	1463775966977
6	1	N	192.168.1.100	1463775967575	1463775967575
7	3	A	192.168.1.100	1463775970534	1463775970534
8	2	A	192.168.1.100	1463775979379	1463775979379

We define $T_{FoundHoneypot}$ as a set of access arrival times for which a message arrives at an active Honeypot. We define $T_{SearchingForHoneypot}$ as a set of time intervals in which the attacker is searching for the active Honeypot.

1) $T_{FoundHoneypot}$ = {Msg2.ArrivalTime, Msg4.ArrivalTime, Msg7.ArrivalTime, Msg8.ArrivalTime}

2) $T_{SearchingForHoneypot}$ = {[Msg1.ArrivalTime, Msg2.ArrivalTime], [Msg3.ArrivalTime, Msg4.ArrivalTime], [Msg5.ArrivalTime, Msg7.ArrivalTime]}

We defined sets of times which potentially provide more reconnaissance information than conventional Honeypot solutions

VI. CONCLUSION

We have shown how a football formation can be used to configure a Honeypot environment to gather information about cyber-attacks. We have also provided a proof of concept experiment to show that our approach is feasible. Our novel approach can be used to gather valuable reconnaissance information about single and ultimately coordinated attacks using well established football plays.

Future research will show, how organizations may use these sets of times to either prevent attacks and/or catch attackers. We further propose that we can use plays from other sports in a Honeypot environment.

ACKNOWLEDGEMENT

The authors would like to thank Lt. J. Bernard Brewton for his invaluable help in this paper. The authors would also like to thank Dr. Frank Li, Dr. Jerome Lewis, and the Division of Mathematics and Computer Science for the use of their Networking Lab.

REFERENCES

[1] Cosmell, H. (2011). 9 Football Formations Every Man Should Know. Retrieved February 18, 2016, from http://www.totalprosports.com/2011/07/26/9-football-formations-every-man-should-know/

[2] Mokube, I., & Adams, M. (2007). *Honeypots: Concepts, Approaches, and Challenges.* North Carolina: Winston-Salem.

[3] Schneier, B. (2000). *Secrets and Lies: Digital security in a networked world.* New York: John Wiley & Sons.

[4] Spitzner, L. (2002). *Honeypots: Tracking Hackers.* Addison-Wesley Professional.

[5] Whitman, M. E., & Mattord, H. J. (2011). *Principles of Information Security.* Cengage Learning.

Detecting and Preventing Information Security Breaches

Haydar Teymourlouei, Lethia Jackson
Department of Computer Science
Bowie State University,
Bowie, MD, USA

Abstract - *Over the last decade, data breaches are constantly growing and have continued to increase globally as new technology continues to evolve. It becomes a lot easier for hackers to breach a system since technology evolved. How can a user secure their data from being breached? Most users are not aware of the proper security methods to protect their personal information. There are thousands of applications and networks which contain sensitive information ranging from bank accounts, passwords, social security numbers, and more. Data breaches will continue to impact institutions worldwide. The importance of protecting data is becoming more prominent since data is compromised daily. Organizations are forced to spend excessive amounts of money on various software solutions to combat intrusion.*

Keywords: Data breach, vulnerability, prevention, monitoring, logs, threat

1 Introduction

As technology is rapidly evolving and becoming a more integral part of the workplace there has been an increase in data breach incidents. Legislation has required companies to have a disaster plan in place in case of a data breach. Additionally, companies are required to report data breaches to central government agencies. Currently thirty-three (33) states have laws requiring public disclosure of security breaches containing sensitive personal information [1]. Many people still believe there is an inability to handle large scale data breach incidents despite new advancements in technology. The public is getting more concerned as the number of data breaches involves high-profiles. Cyber criminal's target organizations (such as business and government) that involve millions of clientele directly or indirectly. There is a dominoe effect when organizations are connected via the network. Once vulnerability affects one organization within the network, then another organization will get infected as well. An example, of this is the Apple iCloud breach where hackers used information from Amazon customer services to get pass security questions on Apple's network.

As security breaches have been increasing over the years, so is the awareness to the public because many state laws require individuals whose personal information is exposed be notified of the breach. Companies and organization use plenty of resources and time to protect their data from outsiders. The most common type of data breach within any particular organization is an insider threat.

To prevent shared resources data breaches there are numerous ways that include educate, encryption, intrusion detection and prevention, filter content, perform vulnerability assessments, patch management, system monitoring, and backup. You would educate the users, by training them on security awareness; it will help them notice odd behavior on the system. I would recommend encryption, because it will better secure information that would be wanted by others. Deploy intrusion detection and protection should be used for systems that are accessible to the Internet. You should content filer, because malicious or compromised Web sites that contain malware and virus that can be downloaded by clicking link.

2 What Is A Data Breach

A data breach is an incident where confidential data is accessed illegally from a hacker. Data breaches may involve personally identifiable information, financial information, intellectual property, trade secrets, or any other information that may either be important to the organization or confidential to its personnel or customers. In accordance to Margaret Rouse of tech target, a data security breach is an incident in which sensitive, protected or confidential data has potentially been viewed, stolen or used by an individual unauthorized to do so [2].

There are many events that can take place to cause a data breach. There are two kinds of breaches, internal and external. Internal data breaches happen when an inexperienced user unknowingly downloads corrupted software, or plugs in a device that is already corrupted ultimately infecting the entire network. External data breaches are when hackers break into a supposedly secure network to either steal information or damage the function of the network. The Government reported 61,000 breaches in 2014. One of the most common perceptions of a data breach is to steal sensitive information. Organizations have started taking precautionary steps. One example of a precautionary step used as a security measure is the insertion of EMV chips in debit cards. Data breaches have the potential to be very damaging as they present a constant attack by hackers to gain access to sensitive information.

2.1 Examples of a Data Breach

Data breaches can be caused by a variety of circumstances. The most basic cause is a hole in the security system of the network. If a hacker finds this hole the network may be compromised. Some of the more common occurrences happen when device users fail to create a strong password. Weak passwords that are closely related to the user may contain

their name, birthdate, pet name, etc. allows hackers easier access to their accounts. Creating strong passwords extends security. There are other aspects of computing which allow easy access to hackers by not encrypting files or including stronger security policies throughout your network, having weak operating system security, flaws or bugs in the operating system, failure to comply to system updates, and cyber espionage. With the emergence of cloud software, attackers are finding more ways to compromise data and networks. Mobile phones are also vulnerable to data breaches due to applications, such as mobile banking or popular games. For example, an Android Trojan known as 'Acecard' that targets mobile banking applications and messages has resurfaced [3].

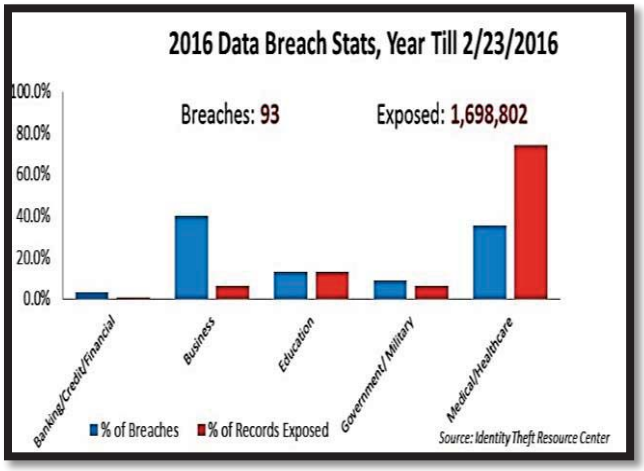


Figure 1: 2016 Data Breach Stats [3]

As shown in the Figure 1 above, reported breaches were placed into one of the following categories: Financial Service Companies, General Businesses, Educational Institutions, Government Agencies and Healthcare Organizations. 1.7 Million Records Already Breached within Just Two Months of 2016 [3]. These data breach incidents were caused either by accident or were intentional. Common data loss methods can be categorized as either: insider theft, hacking, employee error/negligence, accidental web exposure, and physical theft. The most common data that gets compromised is: social security numbers, credit card numbers, email passwords, and protected health information. Data that is compromised commonly affects the individual through incidents such as: financial loss, compromised intellectual property and dwindling customer confidence.

Many of the data breaches are caused by malicious attacks which lead to the loss of critical data. Here is the list of sectors that have been impacted by data breaches during the first two months of 2016. Washington State Health Authority (HCA) on February 2, 2016 notified that 91,000 records of Apple Health (Medicaid) clients were accessed without any authorization by an employee [3]. The information that was comprised is: ssn, dates of birth, client ID numbers and private health information. Healthcare in fact has been the highest entity to report having data breaches while business and educational entities were the next most common type to report a data breach. Specifically, 187 health data breaches accounted for 21.1 percent of the total number of incidents. The financial service industry had the next highest number of breaches with 143, accounting for 16.1 percent of the total. Government was the third highest sector, accounting for 15.8 percent of breaches with 140 incidents total [4]. Also, another data breach incident occurred January 5, 2016 in Southern New Hampshire University. There was a configuration error on a third party vendor which exposed student information such as the student's names, email addresses, IDs, course details etc. According to the report 140,000 students have been affected due to the breach even though the university has only 70,000 enrollments. It is believed that the discrepancy in numbers may mean that both former and current students have been affected [5].

2.2 Causes of Data Breaches

There are many causes of data breaches in the world today. A data breach can occur for many reasons ranging from physical loss such as theft to malicious software attacks. For example, system vulnerabilities, theft, weak security controls, misconfigured access permissions and outdated operating system and applications all leave the system vulnerable. These types of vulnerable provide an easy target for hacking into the system and leaking vital information. It is important to secure sensitive data because of possible internal threats. There are two types of internal threats. One type of internal threat is based on unintentional human error and the other internal threat is intentional. The most common data breach is human error and according to a study conducted by Verizon in 2014, it accounted for 44% of all errors [6].

Employee negligence or human error could also put a system at risk for infiltration from an outside source. There are several examples of unintentional human error. One common example is not having a strong password. When users of a device fail to create a strong password it allows easy access to their accounts. Password security extends to other aspects of computing such as not encrypting files or including stronger security policies throughout your network. A hacker can easily steal or figure out a password if it does not contain a combination of letters, numbers and special characters. Another example of an unintentional human error will occur for instance if a person is not aware of the security protocols and procedures. In this example, an unintentional error occurs when someone sent a sensitive document to the wrong person. Additionally, an unintentional human error could be as simple as the user's hard drive or flash drive gets lost or stolen in which case their private information can be breached. An intentional attack is usually due to either a disgruntled employee and/or the hacker is seeking monetary gain. The biggest threat to a company comes from hackers who are trying to get into the system for financial gain, or personal enjoyment.

2.3 Cost of a Data Breaches

The cost associated with a data breach has escalated due to the increase of incidents as compared to previous years. In today's time every organization or business that collects money and/or personal information is a victim of a data

breach. At the top of the list for the most common entity that receives frequent data breaches is healthcare. Next on the list is education followed by retail and other financial services. The average monthly cost for cleaning and counteractive procedures after a data breach is as high as $20,000 per day. A study by Ponemon implies that the global average cost per data breach has risen from $145 in 2013 to $154 in 2014 which accounts for a 6% increase. The cost includes post data breach procedures including: investigative and remedial actions, setting up hotlines, legal and consultation fees, notifications, incident response unit, etc.… [6].

As stated by law, most states are required to notify outside agencies and law enforcement of a data breach incident. To recover data after the data has been breached is costly as well. After a full investigation of a data breach, an organization's recovery costs could include hiring experienced forensic experts as well as IT experts resulting in loss of the organization's productivity and hence adds to that institution's financial losses.

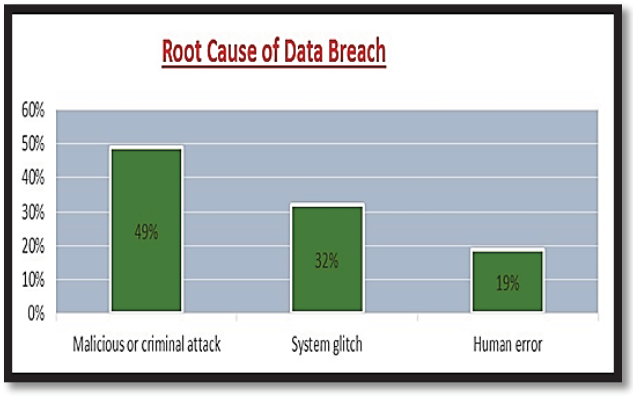


Figure 2: Root Cause of Data Breach [3]

Figure 2 provides the root causes of data breaches. Malicious or criminal attacks are the main root factor for 49% of all data breaches. System glitches which could include both IT and business process failures are 32% of root causes of a data breach. Whereas human error is responsible for 19% of all data breaches.

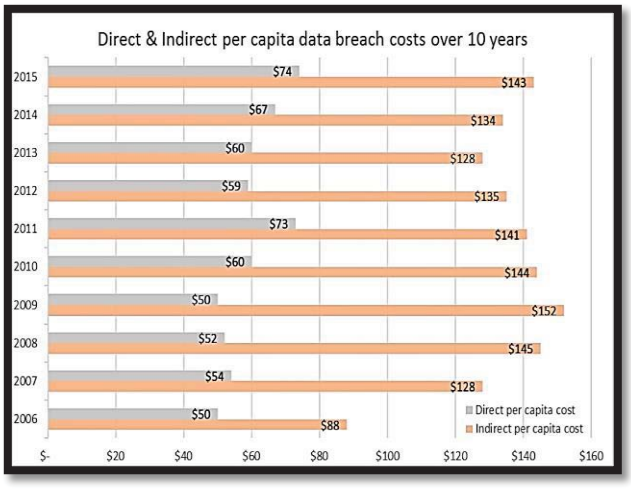


Figure 3: Direct & Indirect Per Capita Data Breach Cost Over 10 Years [3]

Figure 3 shows the cost of a data breach in 2015 and the trends that have impacted it. Direct costs are considered money spent on resources to minimize the consequences of data breach. Whereas, indirect costs is money spent on existing internal resources to deal with data breaches. The average cost per lost or stolen record containing sensitive data is $217 for 2015. There has been a substantial increase of $16 per record breached in comparison to year 2014 which is close to an 8% increase. The average cost of $217 consists of $74 towards direct per capita cost and the remaining $143 towards indirect per capita cost [7].

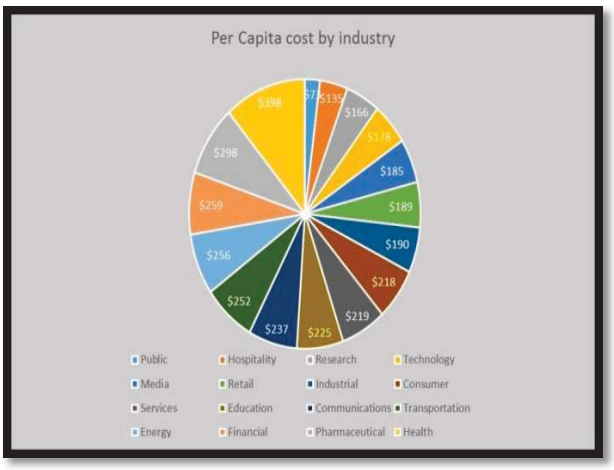


Figure 4: Per Capita Cost by Industry [3]

Figure 4 shows all the sectors that are victims of data breaches and thus healthcare and technology have the highest data breach cost. As shown they tend to have a per capita data breach that cost more than the mean of $217. However, public, hospitality and research have a per capita cost well below the overall mean value.

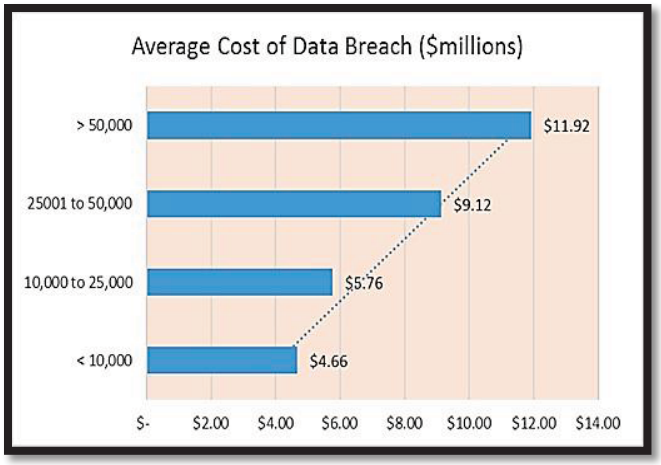


Figure 5: Average Cost of Data Breach ($Millions) [3]

Figure 5 shows the average cost of data breach for 2015. The cost of data breaches increase as the number of lost records increase. For instance, companies that had data breaches involving less than 10,000 records had an average cost $4.7 million and the companies with a loss of more than

50,000 records had a cost of $11.9 million. The number of breached records per incident in 2015 ranged from 5,655 to 96,550 records. The average number of breached records was 28,070 [7]. There are numerous factors that contribute to the increase of lost business costs such as: legal services, investigation & forensics, increased customer acquisition activities and diminished goodwill. Businesses need to make practical decisions and invest in cautious strategies to reduce the cost of data breaches. For example, investing in employee training and incident/data loss prevention procedures and plans.

3 What is an Insider Threat?

Employees getting unauthorized access to an organization's system or information either intentionally or accidentally. Insider threats could also arise by having others gain access to critical data without the required authorization [8]. Insider threats have been a growing concern for organizations mainly because they are not able to deter insider threats in a timely manner. It presents a challenge for companies and organizations who have limited IT resources because they do not know where to start. The process is very simple and easy to implement and manage, and does not require a dedicated resource. For instance, develop and implement methods to prevent unauthorized access to data such as monitoring and limiting the access to critical data. Companies can establish systems that use intelligence and analysis to recognize unauthorized access to prevent accidental access which can help in early detection of data breach incidents.

Other simple processes to prevent a data breach is to implement policies that only give access based on user roles as well as install data loss prevention solutions. Organization should limit the ways their employees interact with systems and networks by implementing IT usage policies and technical measures. All of these types of policies focus only on critical data rather than focusing on all data which is redundant and a waste of time and effort. For example an organizational usage policy would restrict employees from connecting their USB storage devices to their workstations. The technical measures would include establishing audit logs to determine if any sensitive data has been deleted, modified, or uploaded by an insider. Another technical measure to prevent critical data from being accessed by an outsider is by having an administrator hide confidential folders/directories so that they are more difficult to find. Additionally, an administrator may edit the permissions of the folder/directory/file so that only certain individual users or groups have certain permissions to read, write, execute, full control, etc.

4 Is Vulnerability Management Necessary?

A network, system or data can never be 100% protected but security measures can be taken to reduce vulnerability and monitor the status of the resources. Beyond Security, a computer security company, states that no single security solution can make a network safe from all attacks [9]. To mitigate vulnerabilities as they are identified, vulnerability management is crucial to stay protected and ahead of attackers. A vulnerability management system will examine networks and devices to identify weaknesses that need to be patched before they are exploited. A vulnerability management is a continuous program that consists of four main higher level processes: discovery, report, prioritization, and response [10]. The process of vulnerability analysis can be done by security analysis software such as network penetration tools or vulnerability scanners.

The detection of vulnerabilities are usually identified by a "White Hat" hacker, an ethical hacker, that tries to discover vulnerabilities before malicious hackers or "Black Hat" hackers find it first. The White Hat will purposely probe to get into the system searching for vulnerabilities. Based on the weaknesses found, the White Hat develops and implements strategies and countermeasures to prevent a real attack. Various types of vulnerability management tools are required since there are many types of malware that have changed over the past decades. Most malware does not target specific systems but instead its' goal is to propagate and randomly infect as many systems as possible based on the system's vulnerabilities.

In 2012, a server located at Connecticut State University was infected with malware and left student's personal information such as credit cards and social security numbers vulnerable before the infection of the malware was discovered. The university did not realize they had sensitive data on the server. Analyzing a system ahead of time and regularly in a vulnerability management strategy can identify important data before it becomes vulnerable. Vulnerability data is only relevant up to the date it was collected and, thus, needs to be continuously updated. It is important to note that over 90% of attacks are carried out through known exploits. In addition, over 80 vulnerabilities are announced each week, and malicious individuals will use these known exploits to get pass other security solutions like firewalls and antivirus programs [11]. The majority of intrusions come from known vulnerabilities. For example, firewalls and intrusion detection systems that watch the perimeters of the network normally do not scan workstations for viruses and malware, while antivirus programs cannot protect data in a database. This indicates that each solution is geared towards a specific area of security, so a balance or a combination of these solutions would help provide optimal security measures.

Emphasis are placed on firewalls and IDS however, vulnerabilities still exist because of the complexity of threats and attack vectors that may come from either the inside, outside or both. A firewall's basic function is to filter network packets or data that comes in and out of the network. It decides whether or not to allow the data to pass based on the system's established rules. This does not prevent, for instance, a disgruntled employee from copying data onto a portable storage device such as a CD or USB flash drive because they are already within the network. An IDS monitor's network activity for suspicious patterns based on known patterns. A sophisticated hacker can bypass IDS if an unknown pattern is used to attack the system. Firewalls and IDS are indeed an essential part of a security system, but vulnerability

management is necessary in identifying security risks that other solutions do not cover.

5 Prevention/Containment Measures For Threats and Breaches

There is a host of measures that can be implemented to prevent security breaches. One of the initial stages of prevention is early detection of security breaches and identification of existing vulnerabilities. Security breaches can become apparent such as when a service unexpectedly shuts down. In other times, security breaches can be ongoing in the background but it will not be detected until the data is compromised. This is why processes should be in place to mitigate identified existing vulnerabilities to prevent future attacks. All software should be kept up to date to reduce the risk of manipulation. Passwords should be changed frequently on a schedule and kept secure. Physical intrusions can be prevented by using keycards or pin codes. Educating employees about data breach vulnerabilities and why these preventative measures are in place is important.

One of the most targeted venues of attacks is the web browser. Every link present on the web browser can potentially lead to an infected page. Configurations that can help keep browsers safer are by disabling JavaScript and Flash plugin when not needed due their inherent security vulnerabilities. To protect privacy and sensitive data, clear the local user data when closing a session to prevent the browser from storing the information. There are various practices to harden and secure network environment. One is patch management which is to keep the operating system and applications updated with the latest patches that typically contain security patches. This is particularly true for known vulnerable applications such as Adobe Flash and Java. Another practice is to enforce a strong password policy such as one that requires a mix of at least eight (8) symbols, letters, and numbers to prevent password cracking. Firewalls and antivirus technologies remain significant in blocking many external attacks and removing malware.

One of the most difficult vulnerabilities to guard against is social engineering because there is no method that can make human behavior fully secure. Countermeasures such as training staff to increase awareness, hiring a security firm, launching anti-social engineering campaigns can be taken. A typical training topic is anti-phishing where a fake email can be sent asking for sensitive information or containing a link to a malicious website. Similar attacks can occur through phone calls asking for sensitive information. There are numerous common vulnerabilities that are easy to fix and costly if exploited including open ports, running services, misconfiguration of firewalls, intrusion detection systems, or system settings. There are free scanning tools such as Netstat that can scan for open ports. The open ports should be mapped to validate applications. Ports should be closed if they are not needed or insecure depending on risk assessment. Similarly, services should be turned off if not in use by the organization. Configuration setting of security components should routinely be reviewed to make sure it is running properly.

5.1 Log Monitoring Can Prevent Information Breach?

Monitoring data can ensure individuals who have access are supposed to have access can track logged-in activities, and prevent a breach as well as manage passwords efficiently. Surveillance is a major function of securing any network and the first deterrent for hacking or infiltrating anything in general. Continuously monitoring surveillance deters the hacker from successfully infiltrating the system. Other prevention methods include regular scans, updated virus protection software and firewalls. There are free software solutions like Barracuda's Firewall as well as paid software solutions like McAfee's Advanced Protection or Patrol that monitor and protect against security breaches.

All operating systems come equipped with a standard log in feature and it is often overlooked. Software that may be used to monitor the system for security has a built-in event logger found in Windows based operating systems. This tool can give administrators (audit on Windows Server) the ability to view events that vary from logins (fails/success), actions against files (access, deletion, creation), and much more. These log files keep track of events that are produced from hardware and software operations. They can be range in notification from informational events to warnings and then to critical errors. However, not all events are collected by default, important audit settings must be turned on in domain group policy and on the folder or device that contains valuable data in order to receive such events. Log files can alert an administrator that a significant file has been modified or deleted or an unsuccessful attempt to do so can be raise a red flag.

In Windows, auditing events go the security log which contains a hidden treasure of security information that can help detect breaches. It is recommended that log file monitoring software should be used to parse log files due to the numerous amount of information contained in log files. Searching through them to produce and audit report can be an overwhelming task because of the sheer number of records. Log file monitoring software presents the parsed files into a clear report. As described previously, log files are key to forensic analysis to determine what is currently happening that may lead to a breach or what happened during a breach. A built-in tool in windows called Event Viewer is where the majority of log files are collected including the security log. Event viewer can be established so that notifications such as email alerts to administrators if a serious error occurs. This allows the administrator to respond in a timely manner. Event viewer log files can be processed with scripts in a computer language of choice. The aim is to feed the script or small application to the log files in an acceptable format so that the application can read directly from the event viewer and create an output that condenses the information found in the log to a clearer format.

The input file is a tab delimited text file with all the log events but it has too much information. This input file is created by using a filtering function within event viewer to collect the specified events such as file or folder deletion as shown in Table 1. It shows a snippet of the source code from the application in the C++ computer language. The output file the same information but in a much more condensed format that is easier to read as shown in Table 2. This table shows sample log file objects access level.

```
while (read)
{
     if((StringInput == "Audit") &&
     (StringInput2 == "Success") && (tab == '\t'))
     {
          // allocated memory
          newNode = new EventNodeStruct;

          // initialize all variables to Unavailable
          newNode-> UserNameID       =  MessageEmpty;
          newNode-> MyDomain         =  MessageEmpty;
          newNode-> IPAddress        =  MessageEmpty;
          newNode-> MyDomain         =  MessageEmpty;
          newNode-> CurrentTime      =  MessageEmpty;
          newNode-> CurrentAM_PM     =  MessageEmpty;
          newNode-> EventID          =  MessageEmpty;
          newNode-> Category         =  MessageEmpty;
          newNode-> Message          =  MessageEmpty;
          newNode-> Object           =  MessageEmpty;
          newNode-> Access           =  MessageEmpty;
          newNode-> SharedrivePath   =  MessageEmpty;

          read >> StringInput; // read date
          newNode ->CurrentDate = StringInput;
          read >> StringInput; // read time
          newNode ->CurrentTime = StringInput;
          read >> StringInput; // read time for AM or PM
          newNode ->CurrentAM_PM = StringInput;
          read >> StringInput; // read space
          read >> StringInput; // read event ID
          newNode ->EventID = StringInput;
          cout << newNode ->EventID <<endl;
          cout << newNode ->EventID <<endl;

          // A network share object was accessed.
          if (newNode ->EventID == "5140")
          {
               read >> StringInput; // read Category
               newNode ->Category = StringInput;
               read >> StringInput;
               newNode ->Category.append(" "+StringInput);
               getline(read,newNode->Message);
               newNode->Message.erase (0,2);

               read >> StringInput;
               while(StringInput != "Account")
                    read >> StringInput;
```

Table 1: Sample Code

	EventID	Category	Date	Time	Message	Name	Domain	IP Address/Location	Object
2	4656	File System	3/11/2016	2:25:16	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_4_4\pj_4_4.Debug	SYNCHRONIZE
3	4656	File System	3/11/2016	2:25:16	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_4_4\pj_4_4	SYNCHRONIZE
4	4656	File System	3/11/2016	2:25:16	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_4_4.Debug	SYNCHRONIZE
5	4656	File System	3/11/2016	2:25:16	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_4_4	SYNCHRONIZE
6	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112	SYNCHRONIZE
7	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem	SYNCHRONIZE
8	4663	File System	3/11/2016	2:25:15	An attempt was made to access.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2	DELETE
9	4663	File System	3/11/2016	2:25:15	An attempt was made to access.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2	DELETE
10	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2.Source.cpp	DELETE
11	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2\pj_2.vcxproj	DELETE
12	4663	File System	3/11/2016	2:25:15	An attempt was made to access.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2	DELETE
13	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2.idb	DELETE
14	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2.ilf	DELETE
15	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2.opensdf	DELETE
16	4663	File System	3/11/2016	2:25:15	An attempt was made to access.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2	DELETE
17	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem	READ_CONTROL
18	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2	SYNCHRONIZE
19	4656	File System	3/11/2016	2:25:15	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2	SYNCHRONIZE
20	4656	File System	3/11/2016	2:25:10	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_4_3\pj_4_3	SYNCHRONIZE
21	4656	File System	3/11/2016	2:25:04	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2\pj2_output.txt	DELETE
22	4656	File System	3/11/2016	2:25:04	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem	READ_CONTROL
23	4656	File System	3/11/2016	2:25:04	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem	SYNCHRONIZE
24	4656	File System	3/11/2016	2:24:48	A handle to an object was requested.	Sregem	DomainName	H:\Homes\Sregem\COSC112\pj_2\pj_2.v11.suo	READ_CONTROL

Table 2: Output Result of Log Files

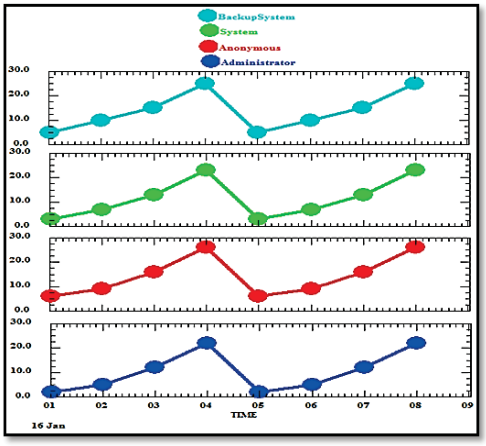


Figure 6: Data Accessed By Users

This plot show how many times each user access protect data from system.

5.2 Preventing And Detecting Security Vulnerabilities Using Software

How can the data be protected against the threat of spyware or insider negligence? In past couple of years there have been many reports of personal data being exposed through theft of laptops, backup drives as well as data being breached when transmitted across networks by unauthorized users. One of the proven techniques to prevent data from getting breached is to encrypt the data. Encryption protects data as the decryption key is requires to unencrypt the data. Without the decryption key unauthorized interceptor cannot access data.

Data protection is very vital to avoid any kind of loss whether the breach is intentional or just human error. Even with all of the security breaches that can occur, there are multiple ways to protect yourself and your organization from a security breach. There are many ways to solidify your network and or data to monitor events and outright prevent them from even occurring. There exists a wide array of third party software solutions that can be used to prevent and detect security vulnerabilities. FileAudit is a commercial solution that offers a simple user interface as shown in Figure 7 and uses audit information to detect file access on servers. It features real time access monitoring so that modifications are reported immediately. This solution mainly track read write and delete accesses in the system. Additionally, it can track failed access attempts, file ownership changes, and file permission modification.

FileAudit also features filtering capabilities to condense audit information by attributes such as: username, domain, date range, event source, access type and status. This solution is non-IT user friendly, meaning managers or personnel that do not have administrative privileges but can better understand and identify important files are able to implement and view audits independently and securely. Special accounts can be created with no administrative privileges but can use the FileAudit solution. Other benefits of FileAudit include centralized alerts, reports, and archiving.

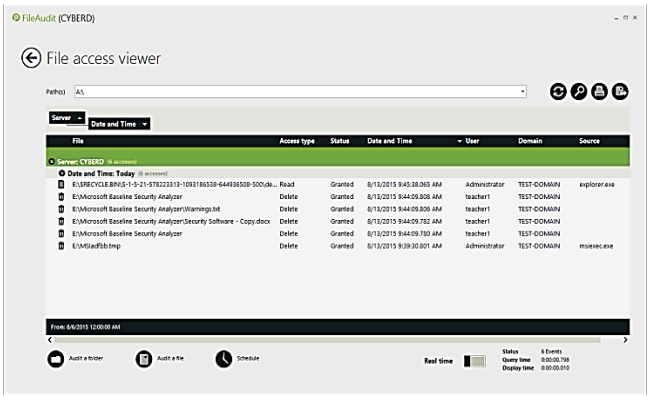


Figure 7: FileAudit Software

Microsoft Baseline Security Analyzer (MBSA) is a free software tool that can perform local or remote scans on windows desktops and servers identify any missing service

packs, security patches, and common security misconfigurations as shown in Figure 8. It features the ability to scan multiple machines and afterwards compiles a report of the state of security patches for each machine. This tool helps find vulnerabilities that can be easily overlooked. However, it should be coupled with other vulnerability scanners such as Tripwire's SecureCheq because by itself it misses certain vulnerabilities.

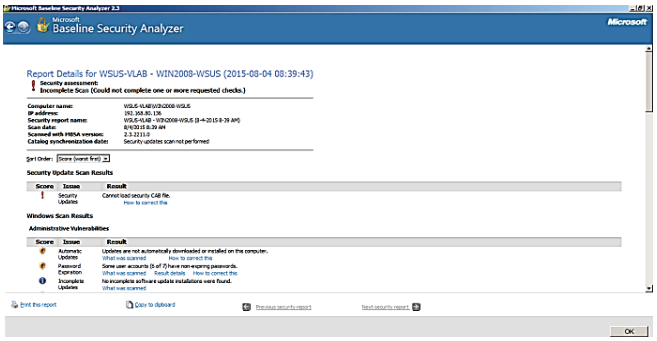


Figure 8: Microsoft Baseline Security Analyzer Software

Metasploit is an open source penetration testing framework that has various tools which scans the network for security vulnerabilities such as open ports and performs penetration tests to access network security. It has a web-based graphical user interface that is easy to navigate as shown in Figure 10. There are a limited number of penetration tests available in a free version that includes useful built-in scanning tools such as Nmap which does port scanning and OS fingerprinting. This tool detects open ports and identifies host and the operating systems on the responsive hosts. The community edition is the free version that provides a way to see all network devices in order to map the network. This also helps detect rogue devices by having a list of connected devices. Metasploit supports importing data from other vulnerability scanners such as Nessus and Nexpose, which can be executed within the Metasploit Community Edition.

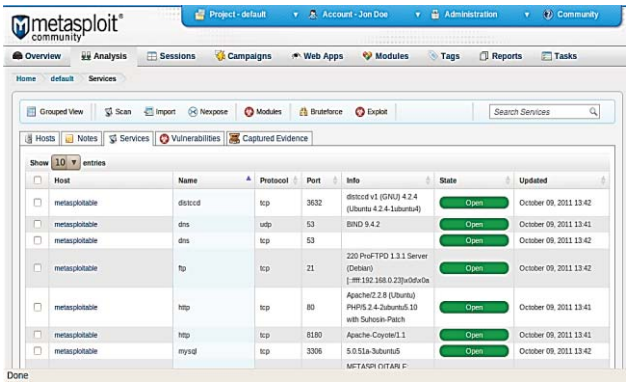


Figure 9: Metasploit Software

6 Conclusions

Data breaches pose a substantial risk of identity theft to critical data that is exposed, it is critical that these breaches get notified as they occur. This provides an opportunity to take appropriate action to reduce the chances of harm should identity theft occur. There are multiple layers of security that can help fortify a network system. Best security practices should be deployed on border security solutions such as firewalls and intrusion detection systems. Internally, antivirus software and vulnerability scanners should protect the connected devices and harden the network. Additionally, auditing solutions should be placed in the network to check for file access and other security events. This allows for detection of successful or unsuccessful intrusion attempts on data.

Security breaches are widespread today as seen in the news. Organizations need to be current with the latest vulnerabilities to prevent known attacks. The importance of continuous vulnerability management should not be overlooked because exploits and viruses are constantly evolving. Although cyber criminals will always pursue weaknesses in computer systems, the countermeasures put in place today will help deter attacks in the future.

7 References

[1] Greenberg, P. (2016, January 4). Security Breach Notification Laws. Retrieved from http://www.ncsl.org/research/telecommunications-and-information-technology/security-breach-notification-laws.aspx

[2] Rouse, M. (2010, May). Data Breach. Retrieved from http://searchsecurity.techtarget.com/definition/data-breach

[3] Garg, R. (2015, October 12). Proactive Measures Go a Long Way in Timely Prevention of Data Loss. Retrieved from http://zecurion.com/blog/

[4] Snell, E. (2015, September 9). Health Data Breaches Account for 21% of Total Incidents. Retrieved from http://healthitsecurity.com/news/health-data-breaches-account-for-21-of-total-incidents

[5] Garg, R. (2015, October 12). Proactive Measures Go a Long Way in Timely Prevention of Data Loss. Retrieved from http://zecurion.com/blog/

[6] Garg, R. (2015, October 7). Can You Really Risk a Data Breach? Retrieved from http://zecurion.com/2015/10/

[7] Garg, R. (2015, October 12). Proactive Measures Go a Long Way in Timely Prevention of Data Loss. Retrieved from http://zecurion.com/blog/

[8] Garg, R. (2016, February 15). Insider Threat Mitigation – A Simple, Secure & Successful Approach. Retrieved from http://zecurion.com/2016/02/15/insider-threat-mitigation-a-simple-secure-successful-approach/

[9] Web Security, Your Site and Your Network. (n.d.). Retrieved March 20, 2016, from http://www.beyondsecurity.com/web-security-and-web-scanning.html

[10] What is Vulnerability Management Anyway? (2013, May 8). Retrieved from http://www.tripwire.com/state-of-security/vulnerability-management/what-is-vulnerability-management-anyway/

[11] Web Application Firewall. (n.d.). Retrieved March 20, 2016, from http://www.applicure.com/solutions/web-application-firewall

SESSION

CRYPTOGRAPHIC TECHNOLOGIES II

Chair(s)

Dr. Kazi Zunnurhain
Dr. Jue-Sam Chou

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

A New Group-based Secret Function Sharing with Variate Threshold

Anneke Soraya Hidayat†, Dae-Soo Kim†, Eun-Jun Yoon‡ and Kee-Young Yoo†*
† School of Computer Science and Engineering
Kyungpook National University
Daegu, South Korea
‡ Department of Cyber Security
Kyungil University
Gyenggsan-Si, Gyeongbuk, South Korea
Email: annekesoraya@gmail.com, stairways@infosec.knu.ac,kr, ejyoon@kiu.kr, yook@knu.ac.kr*
*Corresponding Author

***Abstract*—Secret sharing is a cryptographic scheme that divides a secret key s to n participants with threshold t. Thus, assuring that no less than t shares can reconstruct the secret. The application of secret sharing also widens not only one group but several hierarchical structure. However, the group-based secret sharing proposed by Liu et al. in 2014 has some drawbacks, which are having an equal threshold for each group and also reconstruct only partial secret when the present groups are less. In this paper, we attempt to apply a group-based hierarchical structure in secret sharing by applying derivation. The proposed scheme is the first approach of combining a hierarchical structure and a group-based secret sharing. The proposed scheme shows a better security than Liu et al. considering the structure of the group. Also, it has variated threshold in each group, which is possible for giving less threshold for high-level participants and more threshold for low-level participants.**

***Keywords*—Secret Sharing, Group Access Control, Hierarchical, Variate Threshold**

***Track*—Cryptographic Technologies, Secret Sharing.**

I. Introduction

Secret sharing has been widely known for securing the key among a group of parties. In the traditional symmetric cryptography algorithms, such as DES [1] and AES [2], a single key is required to encrypt and decrypt the secret message among two parties. The same concept is also applied to asymmetric cryptography [3]. The problem arises when the number of parties is increased to more than two. As the problem considered in Shamirs paper [4], we need a numerous number of keys to be shared among parties in one group for securing the documents in a cabinet, thus they can combine the keys and open the locks of the cabinet by the keys carried by the given parties. However, the number of keys and locks increases exponentially and is impossible to apply in the practical world.

In the threshold cryptography, secret sharing has been widely known for securing the key among a group of parties. Secret sharing presented as $(t, n)-$threshold in which at least t number of parties among the n parties in one particular group can reconstruct the secret key s. This system reduces the computational time required and simplifies the system. One of the world practical examples of secret sharing is its application in DNSSec [5], [6].

Secret sharing method was introduced by Shamir [4] and Blakley [7] in 1979. Both of them proposed secret sharing with different approaches, polynomial interpolation and geometry spaces, respectively. In Shamirs scheme, secret sharing divides a secret key s into n different pieces of shares by using a polynomial with $t-1$ degree. The reconstruction phase is conducted by combining, at least, t shares from the group and calculating altogether by using Lagrange interpolation. These schemes assure that no less than t shares can reconstruct the secret key. Shamirs secret sharing is considered safe since the third party (active adversary in this context) cannot break the computational bound, in other words, this scheme is unconditionally secure. Apart from the mentioned two schemes, there are some secret sharing methods which is use different mathematical approaches [8], [9].

In the wide 30 years of secret sharing innovation, several approaches have been proposed such as Chinese Remainder Theorem based [8], [9], Boolean operation based [10]. Other secret sharing [11] which is based on the polynomial differential-based. Blundo [12] and Pang [13], which based on multi-secret sharing scheme.

The hierarchical secret sharing was first introduced by Tassa [14] in 2007. It used Birkhoff interpolation which is applied to the hierarchical system in secret sharing. Then, more schemes related to hierarchical sharing has been proposed [15]. Suppose there is a set of participants that is partitioned into several levels, where each group has threshold relation as $t_1 > t_0$. Thus, by combining shares from each group by the given threshold condition is possible.

The group-scalable secret image sharing scheme was proposed by Liu et al. in 2014 [16] . The secret image is divided to four different quality image and distributed to each group by applying secret sharing polynomial. However, this paper can only reconstruct partial secret when there is less group presence, also, the threshold in each group remains the same.

In this paper, we propose a new group-based secret function sharing with variate threshold which is the first scheme with

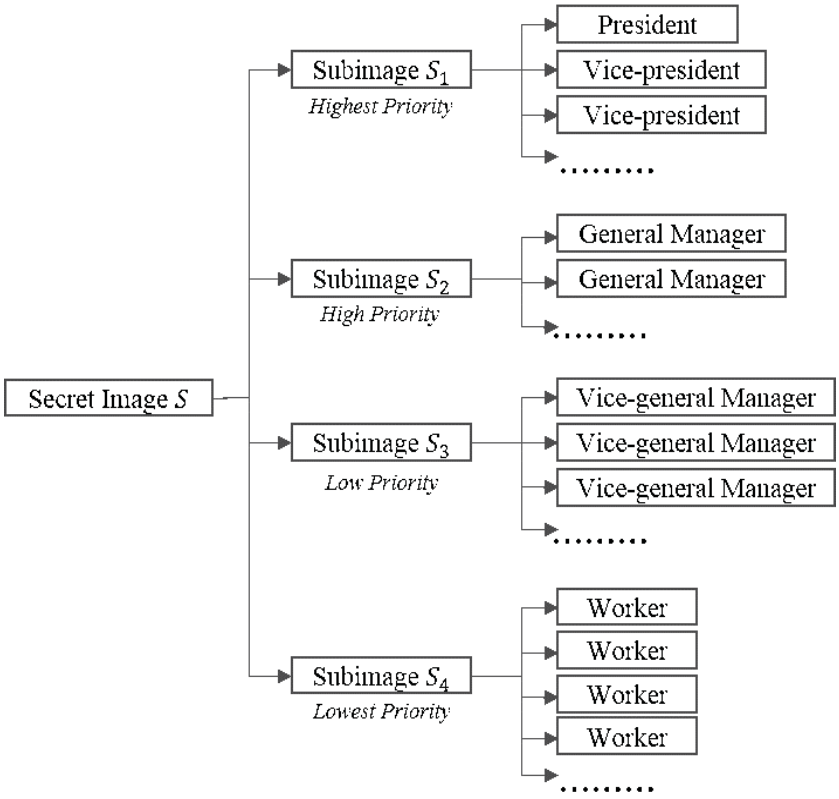


Fig. 1. Hierarchical structure for several management groups

hierarchical and group-based secret sharing. This scheme is also an improvement from previous schemes [11], [17] which based on the polynomial differential-based. Also, we are using a secret function $g(x)$ with degree k as the secret polynomial key. The variate threshold here means that this scheme can produce a different variance of threshold in several groups, which are bound to each other. Our intention is to improve the basic Shamirs secret sharing with Lagrange interpolation rather than using another mathematical approach, such as Birkhoff interpolation. We prove that it has a higher security compared to Liu et al.'s scheme.

This paper is organized as follows. Section two shows the related works. The proposed scheme is described in section three. The comparison of performances between previous works and proposed scheme is discussed in section four. Finally, some conclusions are given in section five.

II. Related Works

A. Shamir's secret sharing

Shamirs secret sharing [4] is based on linear polynomial interpolation. Suppose the dealer $\mathcal{D}$ ought to divide the secret key s to n participants with the threshold value t. The dealer generates polynomial $f(x)$ with degree $t-1$ with s inserted as one of the variables. The polynomial equation shown as below.

$$f(x) = s + a_1 x + ... + a_{t-1} x^{t-1} \mod q \qquad (1)$$

p is a large prime number, s is the secret message and $a_i \in p, i = 1, \ldots, t-1$ are the random number generated in Z_p. Meanwhile, x is each participants $\mathcal{ID}$. Each participant will receive one unique share $y_j = f(x_j), j = 1, \ldots, n$. Here, $n \leq p$ is suitable to apply, since it may reduce the collision between each participants shares. In the reconstruction phase, the dealer will collect no less than t shares (x_j, y_j) from a group of n participants, and conduct the computation as follow.

$$f(x) = \sum_{j=1}^{m} y_j \prod_{k=1, k\neq j}^{m} \frac{x - x_k}{x_j - x_k} \mod q \qquad (2)$$

Equation 2 is used to reconstruct the original polynomial $f(x)$. Variable m stands for the number of available participants, where $t \leq m \leq n$. Here, it has to be noted that participants cannot be less than t. Otherwise, the polynomial cannot be reconstructed due to the Lagrange interpolation correctness. Finally, the secret key s is equal to $f(0)$. All transmissions in this scheme between dealer and participants are assumed to be sent via a secure channel. The model of this scheme is centered; there is no interaction between participants.

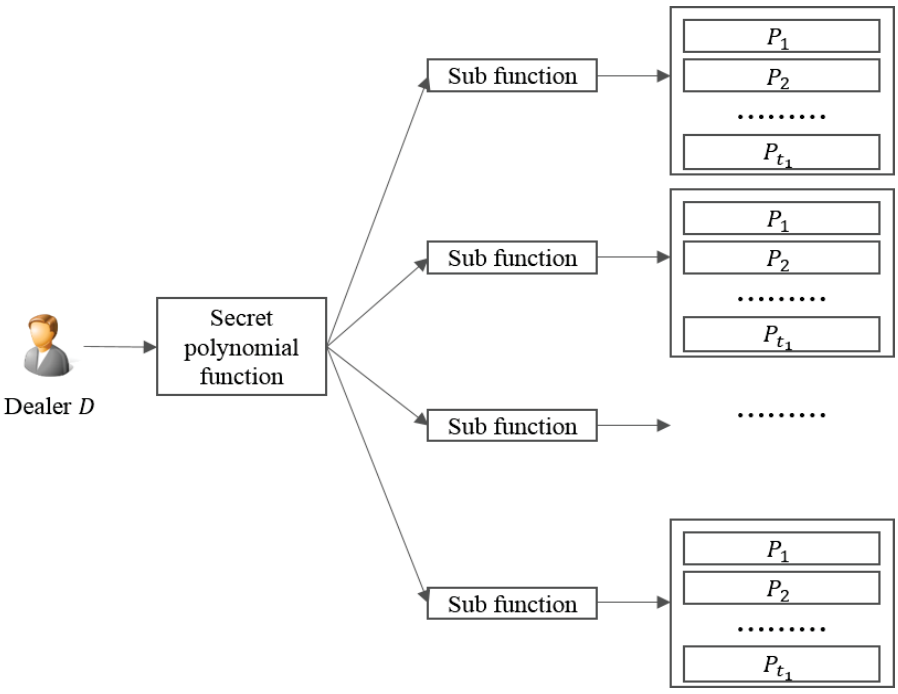


Fig. 2. Hierarchical group structure for distribution phase

B. Group Scalable Secret Image Sharing

Liu et al. proposed a grouped-scalable secret image sharing scheme in 2014 [16]. This scheme used an image as a secret key which is denoted as a secret image as shown in figure 1. The dealer divides the secret image $\mathcal{S}$ into 4 sub-images $S_i, i = \{1, ..., 4\}$by using a bit-plane decomposition by using the expression below.

$$
\begin{aligned}
l_i &= \{b_{i_8} b_{i_7} b_{i_6} b_{i_5} b_{i_4} b_{i_3} b_{i_2} b_{i_1}\} \\
&\downarrow \\
S_1 &= (S_1 || l_{b_8} || l_{b_4}) \\
S_2 &= (S_2 || l_{b_7} || l_{b_3}) \\
S_3 &= (S_3 || l_{b_6} || l_{b_2}) \\
S_4 &= (S_4 || l_{b_5} || l_{b_1})
\end{aligned}
$$

where l_i is the pixel of the secret image $\mathcal{S}$, b_{i_j} is the byte expression of each pixel l_i and $i = 1, \ldots, M \times N$. M and N is the width and height of the image. Four sub-images act as secret keys for each management group and divides each by using secret sharing schemes. Before the shares are distributed to participants in each group, each share hides in a cover image to prevent detection from the third party and disclose the shares into a camouflage.

In the reconstruction phase, it simply reverses the distribution phase on each groups sub-images. Each sub-image has different quality of output, although it shows a lower quality of secret image depending on the priority of the groups. The secret images will show a better quality depending on how many groups are joined. Should any reader wishes to look at the detailed algorithm, please refer to the paper.

This scheme is only limited to four subgroups due to the bit-plane limitation. Practically, every management groups are not always in a static number and can dynamically change. Since the output is an image, each sub-group can somewhat guess the object of the image, without respect to joining with another group.

III. Proposed Scheme

A. Preliminary

The dealer $\mathcal{D}$ generates the polynomial secret function $h(x)$ with degree k.

$$h(x) = a_0 + a_1 x + \ldots + a_k x^k \mod q \tag{3}$$

Here, there are three important variables which have to be set in advance. k, r, and g are the polynomial secret degree, the number of round/integral which will be explained in the next subsection, and the number of total group assigns in the hierarchical structure, respectively. First, the dealer assigns the number of groups g and the secret polynomial degree k. Thus, the number of rounds/integral r are shown using the relationship $r = g - 1$. The number of rounds/integral r purpose is to appoint the number of group.

B. Share Distribution

The dealer $\mathcal{D}$ computes the distribution phase by using the given variables which are assigned in the preliminary stage. The distribution algorithm will be described as follows.

Calculation Example of Distribution Phase

1. Given $g(x) = x^2 + x + 1$, $k = 2$, g = 4, and $r = 3$. All integers are performed in Z_{23}

2. Integral by $r = 3$

$$\iiint g(x) = \frac{1}{60}x^5 + \frac{1}{24}x^4 + \frac{1}{6}x^3 + \frac{C_1}{2}x^2 + C_2x + C_3$$

3. Generate the function for groups by transform the denominators into binary polynomial $GF(2^8)$

$f_0(x) = x^5 + x^4 + x^3 + 2x^2 + 5x + 13$

$f_1(x) = x^5 + x^4 + x^3 + x^2$

$f_2(x) = x^4 + x^3$

$f_3(x) = x^2 + x$

$f_4(x) = x$

4. Calculate each shares for participants for each groups

**Assume that each participant ID $x_i = i, i = \{1, \dots, n\}$*

Group 0	Group 1	Group 2
$f_0(1) = 0$	$f_1(1) = 4$	$f_2(1) = 2$
$f_0(2) = 18$	$f_1(2) = 14$	$f_2(2) = 1$
...	...	...
$f_0(6) = 87$	$f_1(6) = 9$	$f_2(5) = 14$

Group 3	Group 4
$f_3(1) = 2$	$f_4(1) = 1$
$f_3(2) = 6$	$f_4(2) = 2$
$f_3(3) = 12$	

Fig. 3. Calculation example of distribution phase

1) Integrate the secret polynomial $h(x)$ by r rounds. Leave the coefficient's fraction as numerator and denom $\frac{a}{b}$.

$$\begin{aligned} H(x) &= \iiint_r h(x)dx \\ &= C_r + \sum_{k=0}^{t+r} \left(\dots \left(C_1 + \sum_{k=0}^{t} \frac{a_k}{k} x^{k+1} \right) \right)_r \end{aligned} \tag{4}$$

where C_r is the random coefficient in Z_p.

2) Generate g sub-function $f_i(x)$, $i = 0, \dots g$ by changing the form in denominator of each variable a_i in $G(x)$ into binary polynomial $GF(2^p)$, generated g groups.

$$f_i(x) = \{polynomial\}_{GF(2^p)} \leftarrow \{denominator\}_{10}$$

where $i = 1, \dots, g$. Each sub-function $f_i(x)$ generated a variate threshold depends on the denominator value.

3) Assign each sub-function in each group and generate shares by using each participant's $\mathcal{ID}$ from the corresponding group. Here, the highest priority group should be the sub-function which has the lowest threshold degree by $t - 1$ and vice versa.

$$f_{i_j}(\mathcal{ID})$$

4) Send the shares to the corresponding group via the private secure channel.

The example of distribution phase shown in the figure 3 with $C_1 = 2, C_2 = 5$ and $C_3 = 13$.

C. Share Reconstruction

The reconstruction phase requires all the groups to reconstruct the corresponding sub-functions in order to reconstruct secret polynomial $g(x)$. The complete detail of reconstruction algorithm will be described as follows.

1) Collect all shares and reconstruct each groups sub-function $f_{i_j}(x), i = \{0, \dots, g-1\}, j = \{1, \dots, t\}$ by using Lagrange interpolation in equation 2 according to each groups threshold value t.
2) The dealer collects all the sub-functions and changes the form from polynomial into an integer in $\mathbb{Z}_p$.

$$\{denominator\}_{10} \leftarrow \{polynomial\}_{GF(2^q)}$$

3) Assign each integer into denominator in $f_0(x)$, such that $G(x) \leftarrow f_0(x)$.
4) Derive the polynomial $f_0(x)$ r times and get the secret polynomial $g(x)$.

$$g(x) = \frac{d^r}{dx}(G(x))$$

IV. Security Analysis

The Shamirs secret sharing schemes analysis has been proposed recently [18]. The correctness and privacy of Shamirs secret sharing follow the Lagrange interpolation theorem. For every, at least, t distinct values $x_1, ..., x_t$ and any t values $y_1, ..., y_t$, there exists a unique polynomial $f(x)$ of degree which determines the secret. It assures that no $t - 1$ can reconstruct the secret. It also shows that participants $\mathcal{ID}$ x does not depend on the secret . The proposed scheme has an equal security level with Shamirs secret sharing as stated below.

Correctness Secret sharing assures that a pool of t or more participants can reconstruct the secret. It is obvious that a set of t participants carries t points of a polynomial

P with a degree $t-1$, which can be reconstructed by computing $s = P(0)$. Hence, we can conclude that $P(0) = f(0) = s$. Additionally, as mentioned above, the participants $\mathcal{ID}$ does not depend on the secret s. The correlation between $\mathcal{ID}$ and secret is shown in the Lagranges interpolation equation (2) linearly simplified below. The ID depends only with the pool of participants with size t.

$$s = \sum_{k=1}^{t} B_k.y_{i_k}, where B_k = \prod_{i \le j \le t, j \ne k} \frac{x_{i_j}}{x_{i_j} - x_{i_k}}$$

Privacy Any participants group with at most $t-1$ wish to reconstruct the secret by using $y_{i_k}, k = 1, , t-1$, together their shares determines a unique polynomial Pwith degree t-1. The polynomial determines every possible secret where $P(0) = a$ and $P(x_i) = y_i$, for $i = 1, , n$. Thus, the probability is shown as below.

$$Pr[f(x_i)_T = \langle y_{i_k} \rangle_{1 \le k \le t-1}] = \frac{1}{p^{t-1}}$$

The proposed scheme has an equal security level to Shamirs secret sharing. However, the secret form is in the function form and the groups which have variated threshold bind the secret and make it difficult to reconstruct. The advantages of the proposed scheme are shown below.

1) The secret function g has a form of $a_0 + a_1 x + \ldots + a_k x^k$. Each of the variables in the function g has a domain of p since all of the secrets are conducted in modular p. If one of the adversaries $\mathcal{A}$ wants to guess the secret function, the probability of $\mathcal{A}$ successfully guess the function is described below.

$$Pr[g|T] = \frac{1}{p + (k-1)p^2}$$

where, $T = \{y_1, y_2, ..., y_t\}$. It also can be extended of one of the adversaries wishes to reconstruct the secret by using y_{i_k} to put on the pool of k participants, it is nearly impossible to construct, since $Pr\big[f(x_i)_T = \langle y_{i_k} \rangle_{1 \le k \le t-1}\big] = \frac{1}{\{p+(k-1)p^2\}^{t-1}}$. Meanwhile, Shamir's secret sharing domain is only p with the probability $\frac{1}{p}$.

2) The number of groups desired affects the threshold for the first group which acts as the function of the lowest group. Consider a big group $\mathcal{G} = \{t_1, t_2, ..., t_g\}$, the threshold relation for each group can be expressed by $t_1 > t_2 > ... > t_g$ Meanwhile in Liu et al. scheme, the threshold of each group remains the same, which does not represent the hierarchical system.

V. Conclusion

In this paper, the group-based secret function sharing with variate threshold has been proposed. Based on the traditional Shamirs secret sharing, we developed a hierarchical secret sharing system which is constructed in several groups with variate threshold in each group. Here, the secret form used is in a function form in order to increase the security and reduce the chance for an adversary to reconstruct the secret with fake shares, even only with a guess. This system requires all the group together to reconstruct the secret. The security analysis shows that the proposed scheme has a higher security compared to Liu et al. scheme.

In the future, we are trying to improve the combination for each group, allowing only a few groups to reconstruct a part of the secret by using a different approach of polynomial interpolation. Also, unlocking the static threshold, makes it possible for dealers to determine each groups threshold without considering the degree of the secret function.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2015R1A2A2A01006824) and by the Ministry of Education(NRF-2015R1D1A1A01060801).

References

[1] P. FIPS, "46-3: Data encryption standard (des)," *National Institute of Standards and Technology*, vol. 25, no. 10, pp. 1–22, 1999.

[2] J. Daemen and V. Rijmen, "The block cipher rijndael," in *Smart Card Research and Applications*. Springer, 1998, pp. 277–284.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[4] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[5] B. Schneier, "Dnssec root key split among seven people," July 28, 2010. [Online]. Available: https://www.schneier.com/blog/archives/2010/07/dnssec_root_key.html

[6] F. Ljunggren, T. Okubo, R. Lamb, and J. Schlyter, "Dnssec practice statement for the root zone ksk operator." ICANN, 2010.

[7] G. R. Blakley *et al.*, "Safeguarding cryptographic keys," in *Proceedings of the national computer conference*, vol. 48, 1979, pp. 313–317.

[8] M. Mignotte, "How to share a secret," in *Cryptography*. Springer, 1983, pp. 371–375.

[9] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE transactions on information theory*, vol. 29, no. 2, pp. 208–210, 1983.

[10] K.-Y. Chao and J.-C. Lin, "Secret image sharing: a boolean-operations-based approach combining benefits of polynomial-based and fast approaches," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 02, pp. 263–285, 2009.

[11] A. S. Hidayat, E.-J. Yoon, and K.-Y. Yoo, "Polynomial differential-based verifiable secret image sharing," in *Korean Multimedia Conference Proceeding*, vol. 18, 2015, pp. 66–69.

[12] C. Blundo, A. De Santis, G. Di Crescenzo, A. G. Gaggia, and U. Vaccaro, "Multi-secret sharing schemes," in *Advances in CryptologyCRYPTO94*. Springer, 1994, pp. 150–163.

[13] L.-J. Pang and Y.-M. Wang, "A new (t, n) multi-secret sharing scheme based on shamirs secret sharing," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 840–848, 2005.

[14] T. Tassa, "Hierarchical threshold secret sharing," *Journal of Cryptology*, vol. 20, no. 2, pp. 237–264, 2007.

[15] P. S. Kumar, R. R. Kurra, A. N. Tentu, and G. Padmavathi, "Multi-level secret sharing scheme for mobile ad-hoc networks," *International Journal of Advanced Networking and Applications*, vol. 6, no. 2, p. 2253, 2014.

[16] W. Liu, A. Wang, C.-C. Chang, Z. Li, and L. Liu, "A grouped-scalable secret image sharing scheme," *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 7095–7109, 2015.

[17] Q. Al Mahmoud, "Polynomial differential-based strong (n, t, n)-verifiable secret sharing," *Information Security, IET*, vol. 7, no. 4, pp. 312–317, 2013.

[18] A. Beimel, "Secret-sharing schemes: a survey," in *Coding and cryptology*. Springer, 2011, pp. 11–46.

Multilevel Threshold Secret Image Sharing based on the Chinese Remainder Theorem

Rosemary Koikara, Anand Paul and Kee-Young Yoo*
School of Computer Science and Engineering
Kyungpook National University
Daegu, South Korea
Email: rosekoikara@gmail.com, paul.editor@gmail.com, yook@knu.ac.kr*
*Corresponding Author

***Abstract*—(t, n)-threshold secret sharing was first introduced by Shamir and Blakley separately in 1979. Apart from this, there are threshold secret sharing schemes which use the Chinese Remainder Theorem (CRT). The above mentioned are three of the most extensively researched tools used for designing a (t, n)-threshold secret sharing scheme. In this paper we propose a scheme for Multilevel Threshold Secret Image Sharing using the CRT. Multilevel Threshold Secret Sharing (MTSS) is a generalization of the classical secret sharing scheme. In MTSS various participants are classified into levels and the secret is reconstructed from the shares submitted by participants depending on the various levels. Every level has a separate threshold such that a higher level will have a threshold value smaller than that of the threshold of a lower level. Now, participants in each level can reconstruct the secret if the number of shares available is equal to or greater than the threshold of that level. Higher level shares may be used to reconstruct a secret along with lower level shares depending on certain rules. Here, we use Chinese remainder theorem based on Asmuth-Bloom's scheme to perform MTSS in which the secret is an image. The use of Asmuth-Bloom's SS makes this scheme unconditionally secure. Our proposed scheme is the first time the CRT is being used for multilevel threshold secret image sharing.**

***Index Terms*—Multilevel Secret Image Sharing, Secret Image Sharing, Chinese Remainder Theorem.**

***Track*—Cryptographic technolgies, Secret Sharing.**

I. Introduction

Secret sharing divides a secret data into shares that are sometimes also known as secret shadows. The shares generated are then distributed among different participants. Now, the various shares that were previously created are used for the reconstruction of the secret. In 1979, Shamir [1] and Blakley [2] independently proposed two different techniques for (t, n)-threshold secret sharing. The basic idea of a (t, n)-threshold secret sharing scheme is that a secret data is distributed among n participants and only t or more shares are sufficient to reveal the secret. Suppose, $t-1$ or fewer shares are available, it will not be possible to recover the secret.

In 1983, Asmuth-Bloom [3] and Mignotte [4] proposed a (t, n)-threshold secret sharing scheme that used the Chinese Remainder Theorem (CRT). But, Asmuth-Bloom's techniques is proved to be more secure than Mignotte's as it involves more random parameters [5]. These two schemes as pointed out by Kaya et al. [5] cannot prevent a dishonest dealer from distributing inconsistent shares to participants. Kaya et al. then proposed a CRT based VSS (Verifiable Secret Sharing) whose security depends on the RSA assumption. Sarkar et al [6] proposed a CRT-based RSA-threshold cryptography for MANETs using VSS. in 2009. In 2012, Lu et al. [7] proposed a secret key distributed storage scheme based on CRT-VSS and trusted computing for MANET.

Multilevel Threshold Secret Sharing (MTSS), which is a generalization of the classical threshold secret sharing, involves participants who belong to different levels. Each level contains a number of participants. In the MTSS scheme proposed by Simmons [8], all the participants are divided into m levels, L_1, L_2,..., L_m. So here, each level, L_i, $1 \leq i \leq m$, has a subset of participants and a threshold value of t_i. Brickell [9] and Ghodosi et al. [10] both proposed ideal MTSS schemes. The former's scheme was inefficient due to large computation complexities. Ghodosi et al.'s scheme which was based on Shamir's threshold secret sharing scheme worked only for small number of shareholders. In 2009, Lin et al. [11] proposed an ideal MTSS based on a modified version of Shamir's threshold secret sharing scheme. Harr and Fuyou [12] proposed the first MTSS based on Asmuth-Bloom's scheme [3] which is unconditionally secure.

A technique for secret image sharing was first proposed by Thien and Lin [13] in 2002. Their scheme was based on Shamir's secret sharing scheme. This topic has since then been researched extensively [14]–[20]. In secret image sharing, an image is taken as a secret and is divided into n shadow images in such a way that it is enough to have t or more shadow images for reconstructing the secret image. Most of the papers are either based on Shamir's scheme or on Blakley's scheme [19], [20]. In 2006, Meher et al. [15] used CRT to share a secret. But, their approach did not produce a threshold scheme. Also, as reported by Shyu et al. [16] in 2008, certain nature images like sea and forest are not dealt by Meher et al. Shyu et al. [16] then proposed a CRT based technique using Mignotte's secret sharing scheme [4] which overcame the previously mentioned problems. But in case there are equal valued secret neighbour pixels then this scheme uses a PRNG to generate pseudo-random numbers. Hence, the fact that the participants need to know the PRNG function and the seed value used poses a huge disadvantage. In 2009, Asmuth-

Bloom's scheme [3] was adopted by Ulutas et al. [21] for secret image sharing. The advantage of this scheme is that the distribution of the PRNG function or the seed value among the participants is not required. This makes it more secure and efficient than using Mignotte's secret sharing scheme.

In this paper, we adopt Ulutas et al.'s secret image sharing scheme to propose a Multilevel Threshold Secret Image Sharing Scheme. Due to the use of Asmuth et al.'s scheme, we are able to reduce the number of values that need to be distributed. This scheme also enables a use of large number of participants. So far there was no CRT based Multilevel Threshold Secret Image Sharing Scheme in literature.

The rest of the paper is organized as follows. In Section two, the CRT, Asmuth-Bloom's secret sharing scheme and Ulutas et al.'s secret image sharing scheme is introduced. The proposed multilevel secret image sharing scheme using CRT is described in Section three. Experimental results obtained and security analysis of the scheme is shown in Section four. Finally, Section five concludes the paper.

II. Related Work

A. The Chinese Remainder Theorem

From [22], suppose m_1, m_2,..., m_r are pairwise relatively prime positive integers, and suppose a_1, a_2,..., a_r are integers. Then the system of r congruences $x \equiv a_i \pmod{m_i}$ $(1 \leq i \leq r)$ has a unique solution modulo $M = m_1 \times \cdots \times m_r$, which is given in Equation 1.

$$x = \sum_{i=1}^{r} a_i M_i y_i \mod M \tag{1}$$

where $M_i = M/m_i$ and $y_i = {M_i}^{-1} \bmod m_i$, for $1 \leq i \leq r$.

B. Asmuth-Bloom's Secret Sharing Scheme

In the Asmuth-Bloom's secret sharing scheme [3] a secret s is shared among n participants by using modular arithmetic in such a way that any t participants can reconstruct the secret by the CRT. For better understanding we divide the scheme into two phases— the share creation phase and the secret reconstruction phase.

In the *share creation phase* the dealer needs to share a secret s among a group of n participants. To create the shares the following set of operations are performed:

1: Choose a relatively prime set of integers, m_0, m_1,..., m_n such that, m_0 is prime and,

$$m_0 \cdot \prod_{i=0}^{t-2} m_{n-1} < \prod_{i=1}^{t} m_i \mod m_i$$

2: The secret, s, belongs to the set Z_{m_0}.

3: The shares S_i are calculated as, $S_i = (s + y \cdot m_0)$ for all $1 \leq i \leq n$, where y is an arbitrary integer that satisfies the condition, $s + y \cdot m_0 \in Z_{m_1 \ldots m_t}$

In the *secret reconstruction phase*, shares of t participants are required to get s back. We have the following system of congruences.

$$\begin{aligned} x &\equiv S_{i_1} \bmod m_{i_1} \\ x &\equiv S_{i_2} \bmod m_{i_2} \\ &\vdots \\ x &\equiv S_{i_t} \bmod m_{i_t} \end{aligned} \tag{2}$$

Using the Chinese remainder theorem we obtain the unique solution modulo $m_{i_1}, m_{i_2}, \ldots, m_{i_t}$ of the above system. The secret s can be obtained as, $s = x \bmod m_0$.

C. Review of Ulutas et al.'s Secret Image Sharing Technique

Ulutas [21] proposed a technique to share an image among n participants with t as threshold, using Asmuth Bloom's secret sharing scheme.

1) Image Sharing Phase: In this phase n shares of the secret image, $\mathcal{S}$, are created so that they can be distributed among the n participants.

1: A set of n integers, $\{m_0, m_1, m_2, \ldots, m_n\}$ are taken such that,

i) $m_0 < m_1 < ... < m_n < 257$ are in range $[0, 255]$.

ii) Each pair of $\{m_0, m_1, m_2, \ldots, m_n\}$ should be co-prime, i.e., gcd(m_i, m_j) = 1, for $0 \leq i, j \leq n$.

iii) $m_0 \cdot \prod_{i=0}^{t-2} m_{n-i} < \prod_{i=1}^{t} m_i \mod m_i$

2: A pixel from the secret image with gray value, s, is taken sequentially from the secret image.

$$a = \begin{cases} s + r \cdot m_0 & \text{if } s < m_0 \\ s - m_0 + r \cdot m_0 & \text{otherwise} \end{cases} \tag{3}$$

where, r is a random positive integer. When $s < m_0$ then, $u < r < \left(\prod_{i=1}^{t} m_i\right)/m_0 - 1$, otherwise, $0 \leq r \leq u$, where u is a user-specified integer. This solves the problem that arises when the consecutive pixels in the secret image are of the same value.

3: The i^{th} share, S_i is calculated as follows,

$$S_i \equiv a (\bmod m_i), \forall i = 1, 2, \ldots, n. \tag{4}$$

4: The steps (2) and (3) are repeated for all pixels of the secret image, $\mathcal{S}$, until all the pixels are processed.

2) Secret Image Reveal Phase: In this phase, the secret image is reconstructed from shares obtained from t participants.

1: For any t share images we need to reconstruct the secret s. Sequentially take each unused pixels S_i for $i = 1, 2, \ldots, t$. Now apply the Chinese Remainder Theorem based on t pairs, $\{(m_i, S_i), 1 \leq i \leq t\}$ to construct the corresponding value a.

2: We reconstruct the value of r as,

$$r = \left| \frac{a}{m_0} \right| \tag{5}$$

If $r \leq u$, store the m_0 + ($a \bmod m_0$), otherwise store ($a \bmod m_0$).

3: Repeat (1) and (2) until all the pixels of t shares are processed.

Thus, the secret image is reconstructed.

III. Proposed Scheme

The proposed scheme comprises of three phases— the share generation phase, the embedding phase and the secret reconstruction phase. In this multilevel secret image sharing scheme we assume that the participants of a secret image, $\mathcal{S}$, are classified into m levels as, L_1, L_2, ..., L_m, where L_1 is the highest level and L_m is the lowest level. Each level is assigned a separate threshold. The threshold at each level is referred to, t_i, i = 1, 2, ..., m. There are various characteristics for the proposed scheme and they are as follows:

1) The threshold of a higher level is lesser than the threshold of a lower level.
2) For a level L_i, the shadow images belonging to that level or any shadow images belonging to a level higher than L_i can be used to reconstruct the secret.
3) During the secret reconstruction phase, suppose we use shadow images of a particular level, L_i, then the number of shadow images available should be equal to or greater than t_i, else the secret image, $\mathcal{S}$, cannot be reconstructed.

In this paper we assume 8 bit per pixel monochrome secret images (grayscale images). Hence, the images considered here will have 256 shades of gray and the range of the pixel values will be, [0, 255]. Now, we discuss the two phases of the idea proposed in this paper in detail.

A. Secret Image Sharing Phase

1: The dealer selects an integer m_0, such that $127 < m_0 < 257$.

2: Now, for each level, L_i, which has n_i participants, the dealer selects a sequence of n_i pairwise coprime positive integers, $\{m_1^i, m_2^i, m_3^i, \ldots, m_{n_i}^i\}$. Hence, these integers will fulfill the following conditions:

i) $127 < m_1^i, m_2^i, m_3^i, \ldots, m_{n_i}^i < 257$

ii) $m_1^i < m_2^i < \cdots < m_{n_i}^i$

iii) Each pair of $\{m_1^i, m_2^i, m_3^i, \ldots, m_{n_i}^i\}$ is coprime, i.e., gcd(m_p^i, m_q^i) = 1, for $0 \le p, q \le n_i$.

iv) $m_0 \cdot m_{n_i-t_i+2}^i \cdot m_{n_i-t_i+3}^i \cdot \cdots \cdot m_{n_i}^i < m_1^i \cdot m_2^i \cdot \cdots \cdot m_{t_i}^i$. This is the condition that is equivalent to the condition for Asmuth-Bloom's scheme but in case of multilevels.

v) gcd(m_0, m_k^i), $k = 1, 2, 3, \cdots, n_i$.

Here, i = 1, 2, ..., m, where m is the number of security levels, and m_k^i is the public information for participant P_k^i in the level L_i. Here, k corresponds to the shares within a level.

3: A pixel with gray value s is taken from the secret image, $\mathcal{S}$, sequentially.

4: Calculate share for the participant, $\mathcal{P}_k^i$ as.

$$s_k^i = \begin{cases} (s + r_i \cdot m_0) \bmod m_k^i & \text{if } s < m_0 \\ (s - m_0 + r_i \cdot m_0) \bmod m_k^i & \text{otherwise} \end{cases} \quad (6)$$

In Equation 6, r_i is a random positive integer the dealer selects for level L_i such that, $m_{n_i-t_i+2}^i \cdot m_{n_i-t_i+3}^i \cdot \cdots \cdot m_{n_i}^i < s + r_i m_0 < m_1^i \cdot m_2^i \cdot \cdots \cdot m_{t_i}^i$. The value $s + r_i \cdot m_0$ should be in this defined threshold t_i-threshold range, because if this were not the case then the value of $s + r_i \cdot m_0$ will be obtainable using less than t_i shares. The value of r_i should be within the following range, $u < r_i < \left(\prod_{k=1}^{t_i} m_k^i\right)/m_0 - 1$ when, $s < m_0$ otherwise $0 \le r_i < u$ where u is a user-specified integer. This is needed to avoid problems when consecutive pixels are of the same value. (The following steps will enable the use of shares that are being held by participants in higher levels, L_h as a share for a lower level, L_l.)

5: The dealer selects a parameter $m_{k,l}^h$ inorder to enable share, s_k^h of a higher level participant to be used as a share in lower level, L_l ($l \implies$ lower level, $h \implies$ higher level). Here, the value of $m_{k,l}^h$ is chosen such that,

$$m_{t_l}^l < m_{k,l}^h < m_{n_l-t_l+2}^l. \quad (7)$$

6: The dealer now computes $\delta s_{k,l}^h$ using the following equation.

$$\delta s_{k,l}^h = (s + r_l m_0 - s_k^h) \bmod m_{k,l}^h \quad (8)$$

7: Now if one wants to use the share s_k^h in a lower security level, L_l, then the share has to be modified as follows.

$$s_k^h = (s_k^h + \delta s_{k,l}^h) \bmod m_{k,l}^h \quad (9)$$

8: The steps (3) to (7) are performed for all the pixels, s of the secret image $\mathcal{S}$. The share obtained at each iteration is the pixel value of the respective shadow image.

The selected $m_{k,l}^h$ values should be coprime to all the other moduli. The values of all moduli being within the defined range is very vital to the scheme. Failure to do so will make it possible to obtain the secret image from less than prescribed threshold shares or lead to failure to obtain secret image from the threshold or more shares.

Each participant, $\mathcal{P}_k^i$, in a security level, L_i, has one share, s_k^i, which is the private information and the public information is p_k^i. The modification values used for enabling a higher security level share to be used as a share in a lower security level is also made available. For this we have public information p_k^i which is the modulus associated with the level L_i. We also have the share modification values used for using a higher security level share as a share in a lower security level. For that, we have public information, $(\delta s_{k,l}^h, m_{k,l}^h)$, for $h = l+1, l+2, \ldots, m$. Here, a level L_h share is used for a level L_l.

B. Embedding Phase

In this phase, the shares obtained by each participant needs to be embedded in an image, which is called a cover image. Here, the cover images act as the carrier of the shares or shadow images. The shadow images can be left as it is or embedded into a cover image for extra security. There are many techniques using which this embedding can be done. For

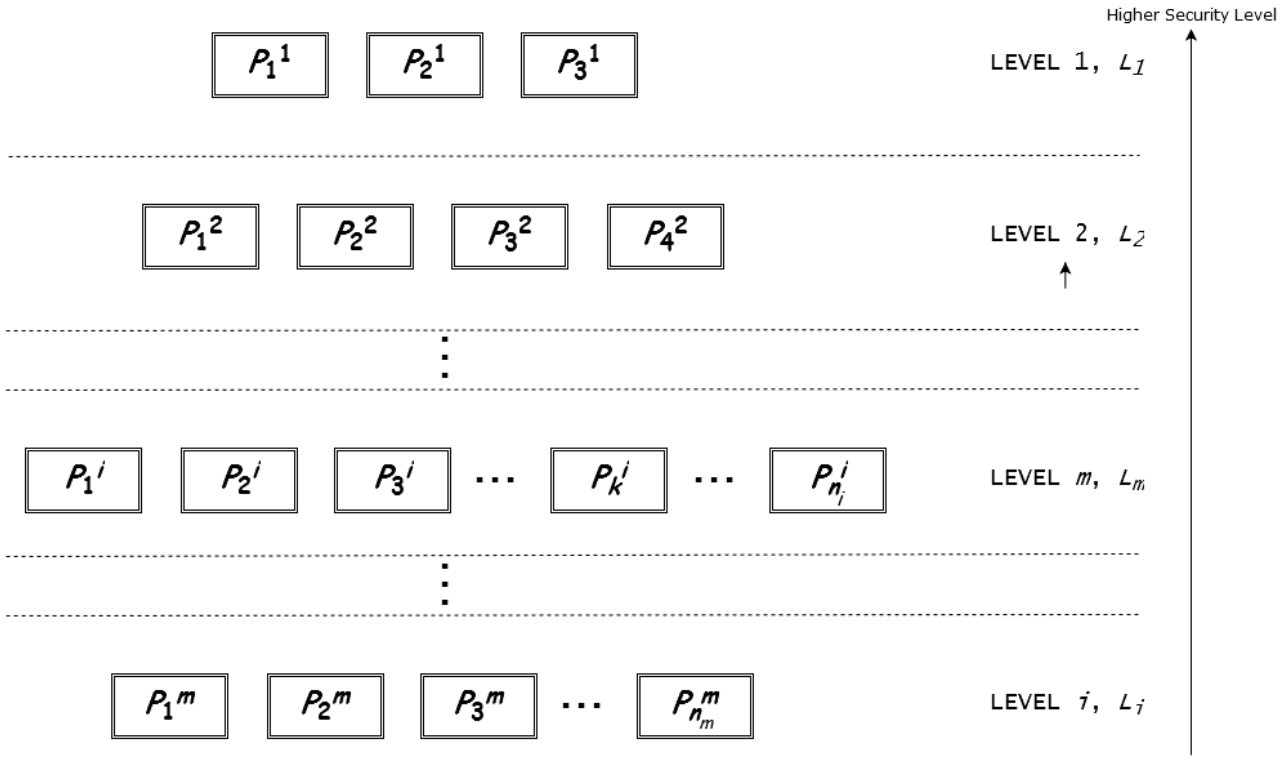


Fig. 1. Multilevel Structure for m Levels

implementation purposes we have done a simple LSB (least significant bit) modification upto a 2^{nd} bit modification [23]. This means that the shares have been embedded into the least significant bits of the pixels of the cover image. This has been done by modifying the last two bits of each of the pixels of the cover image according to the value of the shares. It is these cover images that are being held by the participants.

It can be noted that certain techniques of MTSS produce shares containing large values, hence they need to be pre-processed to come within the [0, 255] pixel range. This is not the case in this proposed scheme.

C. Secret Image Reconstruction

In this phase, the secret has to be reconstructed from the shares submitted to the dealer by the participants. The shares are first extracted from the cover images of the participants. At the secret image sharing phase there were n_i participants at each level L_i. In this phase it is sufficient to have t_i or more shares to reconstruct the secret image where t_i is the threshold at each level L_i. Hence, the secret image can be reconstructed if the number of shares belonging to a level, L_i or a level L_h that has a higher security level than level L_i is more than or equal to the threshold t_i. The detailed description of the algorithm used for the secret image reconstruction proposed in this paper is given below.

1: For a level L_i, we need to reconstruct the secret image such that the number of shares available is more than or equal to t_i. Sequentially, take each unused pixels, a_k for $k = 1, 2, \cdots, t_{n_i}$ from each of the shares from the participants.
2: Apply the Chinese Remainder theorem based on t_{n_i} pairs, $\left\{(m_k^i, a_k), 1 \leq i \leq t_{n_i}\right\}$ to reconstruct the corresponding value of $s' = s + r_i \cdot m_0$.
3: Reconstruct the value of r_i as,

$$r_i = \left| \frac{s'}{m_0} \right| \tag{10}$$

If, $r_i \leq u$, store the $m_0 + (s' \bmod m_0)$, otherwise store $s' \bmod m_0$.
4: Repeat (1) to (3) until all the pixels of t_{n_i} shares are processed.

Suppose, there is a share that belongs to a participant $\mathcal{P}_k^h$ that has a higher security level, then, the share pixel value is modified to $(s_k^h + \delta S_{k,l}^h)$ and $m_{k,l}^h$ is used as the modulus for that share while modifying the share value.

IV. Experimental Results and Analysis

This section describes the experimental results obtained from the implementation performed so far. The proposed method was implemented using MATLAB. We performed the experiments for m = 3, i.e., a system that has 3 security levels. For experimental purposes we assumed that, $n_1 = 4, n_2 = 5, n_3 = 5$ and $t_1 = 2, t_2 = 3, t_3 = 4$. We used fourteen 512×512 pixels gray scale images as cover images for the fourteen participants and a 256×256 pixels grayscale images as a secret images as shown in Fig. 2(a), and Fig. 2(b),

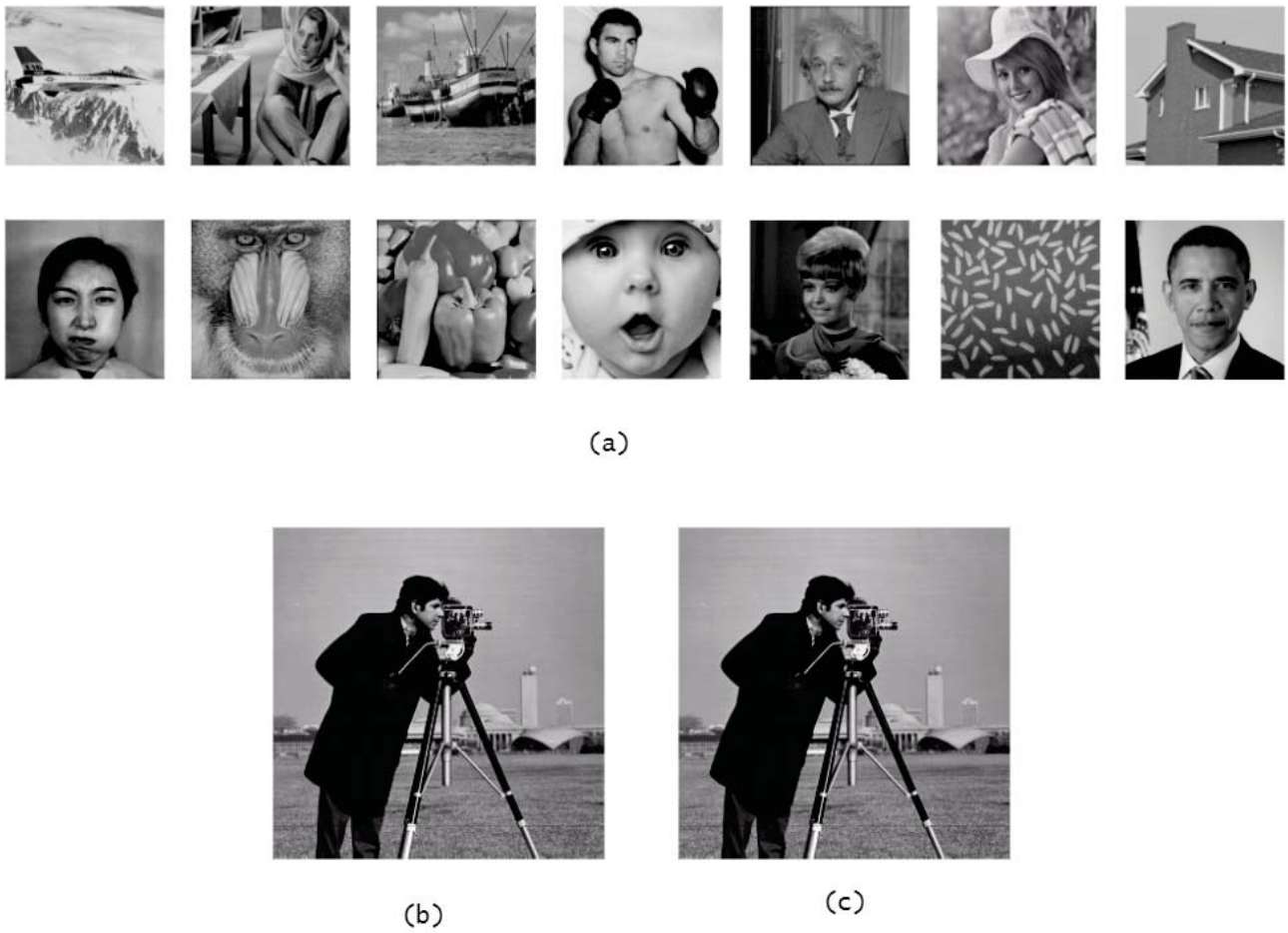


Fig. 2. (a) Cover Images (b) Original Secret Image (c) Reconstructed Secret Image

respectively. We obtain fourteen noise like shares for the participants. The secret image was reconstructed completely as shown in Fig. 2(c).

The user generated u value as used in Ulutas' scheme makes sure that the repetitive pixels in a secret image are randomized. The size of the share obtained is equal to the secret image. This share is then embedded into a cover image to secure it.

A. Analysis

The analysis of this scheme shows us that:

1) t_i or more than t_i are required to recover the secret image else it will not be recovered.
2) The ranges that have been set for the various modulus values ensures that the threshold value is maintained.
3) This is unconditionally secure like Asmuth-Bloom's threshold secret sharing scheme [3] as proved by Kaya et al. [5]

V. Conclusion

In this paper we have introduced a Multilevel Secret Image Sharing Scheme based on the CRT. This is the first time the CRT is being used for Multilevel Threshold Secret Image Sharing. The security of this scheme as mentioned previously is perfect. This scheme introduces a generalization of secret image sharing where the participants are divided into multiple levels and each level is at a different security level when compared to others. On satisfying the threshold conditions, the participants in a level can recover the secret image. It is possible for a higher security level participant to submit its share in order to reconstruct the secret image. In this scheme a participant keeps the modified cover image or shadow image as its share.

This scheme requires a number of public information to be kept with the dealer. We intend to continue this work to reduce the number of public information required. This will reduce the communication payload. Future work will also have to be done to reduce the number of constraints put in this scheme.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01060801) and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP). [No. 10041145, Self-Organized Software platform (SoSp) for Welfare Devices]

REFERENCES

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] G. R. Blakley *et al.*, "Safeguarding cryptographic keys," in *Proceedings of the national computer conference*, vol. 48, 1979, pp. 313–317.

[3] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE transactions on information theory*, vol. 30, no. 2, pp. 208–210, 1983.

[4] M. Mignotte, "How to share a secret," in *Cryptography*. Springer, 1982, pp. 371–375.

[5] K. Kaya and A. A. Selçuk, "A verifiable secret sharing scheme based on the chinese remainder theorem," in *Progress in Cryptology-INDOCRYPT 2008*. Springer, 2008, pp. 414–425.

[6] S. Sarkar, B. Kisku, S. Misra, and M. S. Obaidat, "Chinese remainder theorem-based rsa-threshold cryptography in manet using verifiable secret sharing scheme," in *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2009, pp. 258–262.

[7] Q. Lu, Y. Xiong, W. Huang, X. Gong, and F. Miao, "A distributed ecc-dss authentication scheme based on crt-vss and trusted computing in manet," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012, pp. 656–665.

[8] G. J. Simmons, "How to (really) share a secret," in *Proceedings on Advances in cryptology*. Springer-Verlag New York, Inc., 1990, pp. 390–448.

[9] E. F. Brickell, "Some ideal secret sharing schemes," in *Advances in CryptologyEUROCRYPT89*. Springer, 1989, pp. 468–475.

[10] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini, "Secret sharing in multilevel and compartmented groups," in *Information Security and Privacy*. Springer, 1998, pp. 367–378.

[11] C. Lin, L. Harn, and D. Yea, "Ideal hierarchical (t, n) secret sharing schemes," in *Proceedings of the Fifth International Conference on Information Assurance and Security (IAS09), Xian, China*. Citeseer, 2009.

[12] L. Harn and M. Fuyou, "Multilevel threshold secret sharing based on the chinese remainder theorem," *Information processing letters*, vol. 114, no. 9, pp. 504–509, 2014.

[13] C.-C. Thien and J.-C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[14] C.-C. Chen and C.-C. Chang, "Secret image sharing using quadratic residues," in *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, vol. 1. IEEE, 2007, pp. 515–518.

[15] P. K. Meher and J. C. Patra, "A new approach to secure distributed storage, sharing and dissemination of digital image," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. IEEE, 2006, pp. 4–pp.

[16] S. J. Shyu and Y.-R. Chen, "Threshold secret image sharing by chinese remainder theorem," in *Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE*. IEEE, 2008, pp. 1332–1337.

[17] L. Bai, S. Biswas, and E. P. Blasch, "An estimation approach to extract multimedia information in distributed steganographic images," in *Information Fusion, 2007 10th International Conference on*. IEEE, 2007, pp. 1–6.

[18] S. Alharthi and P. K. Atrey, "An improved scheme for secret image sharing," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1661–1666.

[19] C.-C. Chen, W.-Y. Fu, and C.-C. Chen, "A geometry-based secret image sharing approach." *J. Inf. Sci. Eng.*, vol. 24, no. 5, pp. 1567–1577, 2008.

[20] H.-K. Tso, "Sharing secret images using blakleys concept," *Optical Engineering*, vol. 47, no. 7, pp. 077 001–077 001, 2008.

[21] M. Ulutas, V. V. Nabiyev, and G. Ulutas, "A new secret image sharing technique based on asmuth bloom's scheme," in *Application of Information and Communication Technologies, 2009. AICT 2009. International Conference on*. IEEE, 2009, pp. 1–5.

[22] D. R. Stinson, *Cryptography: theory and practice*. CRC press, 2005.

[23] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[24] M.-H. Tsai and C.-C. Chen, "A study on secret image sharing," in *Proceedings of the 6th International Workshop on Image Media Quality and its Applications, Tokyo, Japan*, 2013.

KERMAN: A Key Establishment Algorithm based on Harvesting Randomness in MANETs

Mohammad Reza Khalili Shoja*, George Traian Amariucai*, Shuangqing Wei† and Jing Deng‡
*Department of Electrical and Computer Engineering, Iowa State University, mkhalili@iastate.edu, gamari@iastate.edu
†School of Electrical Engineering and Computer Science, Louisiana State University, swei@lsu.edu
‡Department of Computer Science of University, North Carolina at Greensboro, jing.deng@uncg.edu

***Abstract*—Secret key agreement between two or multiple devices in a network is usually dependent upon a public-key infrastructure. However, in the cases when no such infrastructure exists, or when the existent infrastructure is not trustworthy, users are left with relatively few methods for establishing secure communication. In this paper, we discuss KERMAN, a secret-common-randomness establishment algorithm for ad-hoc networks, which works by harvesting randomness directly from the network routing metadata, thus achieving both pure randomness generation and (implicitly) secret-key agreement. KERMAN relies on the route discovery phase of an ad-hoc network employing the Dynamic Source Routing protocol. The algorithm is evaluated for various network parameters, and two different levels of complexity, in an OPNET ad-hoc network simulator. Our results show that, in a very short time, thousands of secret random bits can be generated network-wide, between different pairs in a network of fifty users.**

***Keywords*—Ad hoc mesh network, Dynamic source routing, Common randomness, Secret key establishment, Minimum entropy**

I. Introduction

Automatic key establishment between two devices in a network is generally performed by public-key-based algorithms (like Diffie-Hellman [1]). In addition to the key exchange protocol, another aspect of key establishment is to ensure that the newly generated secret key is truly random. While minimum standards for software-based randomness quality are generally being enforced [2], many applications rely on often costly hardware-based *true random generators* [3].

In this paper, we discuss KERMAN –a Key Establishment Algorithm based on Harvesting Randomness in MANETs. KERMAN builds upon the observation that a readily-available source of randomness is the network dynamics. This type of randomness is evident in easily-accessible networking metadata such as traffic loads, packet delays or dropped-packet rates. KERMAN's main focus is on mobile ad-hoc networks (MANETs), and the source of its randomness is routing metadata. A very useful feature of routing metadata is that it can easily be made available to the devices that took part in the routing process, but it is usually unavailable to those devices that were not part of the route. Our algorithm is thus based on establishing *secret common randomness* between two devices in a mobile ad-hoc network. Once established, this common randomness can be further processed into secret keys.

Common randomness was pioneered in [4], [5], [6], where it is shown that if two parties, Alice and Bob, have access to two correlated random variables (RVs) X' and Y' respectively, a secret key can be established between them through public discussions and random-binning-like (e.g. hashing) operations. The key should remain secret from an adversary eavesdropper (Eve) who overhears the public discussions, and possesses side information (in the form of a third RV Z) correlated with that available at Alice and Bob. Common-randomness-based key establishment generally consists of three phases. First, during the *advantage distillation* phase, Alice and Bob have to agree on two other RVs X and Y, such that $H(X|Y) < H(X|Z)$ and $H(Y|X) < H(Y|Z)$, where $H(\cdot)$ is the standard Shannon entropy. Then Alice and Bob (and also Eve) sample their respective random variables many times, producing sequences of values. Second, during the *information reconciliation* phase, Alice and Bob publicly exchange further messages to agree on the same single sequence of values. Third, during the *privacy amplification* phase, because the agreed-upon sequence is not completely unknown to Eve, Alice and Bob run a randomness extractor on it, to produce a secret key (a shorter sequence which, from Eve's perspective, is uniformly distributed over its space). The ideas of [4], [5] have been recently applied to secret key generation in wireless systems, where secure *common randomness* is attained by exploiting reciprocal properties of wireless channels or other auxiliary random sources in the physical layer [7], [8], [9], [10], [11], [12], [13], [14], [15]. It is important to note that, while [4], [5], [6] consider an asymptotic approach, in practice Alice and Bob do not usually have access to large numbers of values drawn from their random variables, but rather to only one or a few values. To address this issue, [16] shows that for such single-shot scenarios, the smooth minimum entropy provides tight upper and lower bounds on the achievable size of the secret key.

In MANETs, the lack of infrastructure, the nodes' mobility and the fact that packets are routed by nodes, instead of fixed devices, have resulted in the need for specialized routing protocols, like the ad-hoc on-demand distance vector AODV routing, or the dynamic source routing (DSR) [17]. For our secret-common-randomness-extraction purposes, DSR appears to be a good candidate. DSR contains two main mechanisms – Route Discovery and Route Maintenance – which work together to establish and maintain routes from senders to receivers. The protocol works with the use of explicit *source routing*, which means that the ordered list of nodes through which a packet will pass is included in the packet header. It is sets of these routing lists that KERMAN processes into secret keys shared between pairs of nodes.

Parts of this work have already been published in [18]. In this paper, we build upon the results in [18], and show how the performance of our key establishment algorithm can be enhanced by employing two different new developments: a heuristic algorithm for set partition and the *spoiling knowledge*

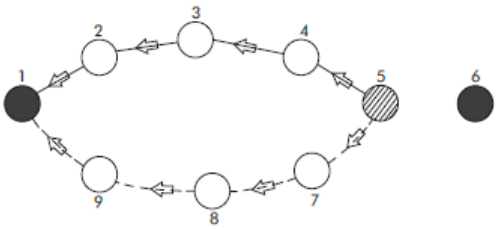


Fig. 1: DSR routing.

technique of [19]. The rest of this paper is organized as follows. Those parts of the DSR protocol that are essential for understanding our algorithm are examined in Section II. In Section III, we describe the system model and state our assumptions. Section IV describes our proposed key establishment algorithm. Simulation results obtained with OPNET Modeler are presented and discussed in Section V, while Section VI draws conclusions and discusses future work.

II. Dynamic Source Routing

Dynamic source routing (DSR) [17] is one of the well-established routing algorithms for ad-hoc networks. Under this protocol, when a user (the sender) decides to send a data packet to a destination, the sender must insert the *source route* in a special position of the packet's header, called the *DSR source route option*. The *source route* is an ordered list of nodes that will help relay the packet from its source to its destination. The sender transmits the packet to the first node in the *source route*. If a node receives a packet for which it is not the final destination, the node will transmit the packet to the next hop indicated by the *source route*, and this process will continue until the packet reaches its destination.

To obtain a suitable source route toward the destination, a sender first searches its own *route cache*. The *route cache* is updated every time a node learns a new valid path through the network (whether or not the node is the source or the destination for that path). If no route is found after searching the route cache, the sender initiates the *route discovery* protocol. During the route discovery, the source and destination become the *initiator* and *target*, respectively.

As a concrete example, suppose node 1 in figure 1 wants to send packets to node 5. Initially, node 1 does not have any route toward node 5, and thus node 1 initiates a route discovery by transmitting a single special local broadcast packet called *route request*. The *route request option* is inserted in the packet's header, following the IP header. To send the route request, the source address of the IP header must be set to the address of the initiator (node 1), while the destination address of IP header must be set to the IP limited broadcast address. These fields must not be changed by the intermediate nodes processing the route request. A node initiating a new route request generates a new identification value for the route request, and places it in the ID field of the route request header. The route request header also contains the address of the initiator and that of the target. The route request ID is meant to differentiate between different requests with the same initiator and target – it should be noted here that the same request may reach an intermediate or destination node twice, over different paths. Each route request header also contains a record listing the address of each intermediate node through which this particular copy of the route request has been forwarded. In our case, the route record initially lists only the address of the initiator node 1. As the packet reaches node 2, this node inserts its own address in the packet's route record, and broadcasts it further, and so on, until the packet reaches the target node 5, at which point its route record contains a valid route (1-2-3-4-5) for transmitting data from node 1 to node 5.

As a general rule, recent route requests received at a node should be recorded in the node's *route request table* – the sufficient information for identifying each request is the tuple (initiator address, target address, route request ID). When a node receives a route request packet, several scenarios can occur. First, if the node is the target, it sends a *route reply* packet to the initiator, and saves a copy of the route (extracted from the route request route record) in a table called the *route cache*. Second, if the node has recently seen another route request message from this same initiator, carrying the same id and target address, or if the node's own address already exists in the route record section of the route request packet (the same request reached the node a second time), this node discards the route request. Third, if the request is new, but the node is not the target, the node inserts its address in the packet's route record, and broadcasts the modified packet. Fourth, if a route exists to the target address in the node's route cache, the node sends the route reply.

In our example in figure 1, node 5 constructs a *route reply* packet and transmits it to the initiator of the route request (node 1). The source address of the IP header of the route reply packet is set to the IP address of sender of route reply (node 5). In our example, node 5 is also the target. But this need not occur. Under the DSR protocol, it is possible that an intermediate node (who is not the target of the route request) already has a path to the target in its route cache. Then it is this node that transmits the route reply back to the initiator, and it is its IP address that gets inserted in the source IP address part of the route reply packet's header. The route reply packet header also contains a *route record*. This route record starts with the address of the first hop after the initiator and ends with the address of the target node (regardless of whether the node that issues the route reply is the target or not). In our example, the route record contained in the route reply packet is (2, 3, 4, 5). Including the address of the initiator node 1 in the route record would be redundant, as the address of node 1 is already included as the destination address in the IP header of the route reply packet. The combination of the route record and destination address in the IP header is the *source route* which the initiator will use for reaching its target. It is also noteworthy that network routes are not always bidirectional. That is, it may not always be possible for node 5 to send its route reply to node 1 using a route obtained by simply inverting the source route. In the more general case, node 5 has to search its own route cache for a route back to node 1. If no such path is found, node 5 should perform its own route discovery for finding a *source route* to node 1.

III. System Model

Mobile ad-hoc networks (MANETs) consist of mobile nodes communicating wirelessly with each other, without pre-existing infrastructure. We consider a *bidirectional* MANET employing dynamic source routing (DSR), in which the nodes are moving in a random fashion in a pre-defined area. The bidirectional network assumption is usually practical, especially when all the nodes in the network belong to the same class of

devices (e.g. smart phones)[1].

According to the route discovery protocol outlined in section II, every node in the network is assumed equally likely to be the initiator of a route request packet, at any given time. Furthermore, we assume that the target of any route request is uniformly distributed among the remaining nodes. Any route discovery instance will return a path through the network (the source route), of a given length. The length of a returned path is distributed according to a probability distribution that depends on all the parameters of the network. Deriving a model for this probability distribution, based on the network parameters, is outside the scope of this work. Hence, in the remainder of this paper, we shall assume that all nodes have access to such an (empirically-derived) probability distribution over the path lengths. That is, if we denote the random variable describing the length of some path r by L_r, then we assume that all the nodes have access to the prior $p(L_r = l)$, for $l = 2, 3, \ldots$. For our experiments, we run our simulation for a long time, and derive $p(L_r = l)$ by counting the paths of equal length. We also assume that *all paths of the same length are equally probable*. To express this notion, denote the random variable that samples a path (or a partial path) by R. Then we can write $p(R = r|L_r = l) = \frac{1}{N_l}$ if the length of path r is l (otherwise the probability is zero), where N_l is the total number of paths of length l. This leads to $p(R = r) = \frac{1}{N_{l_r}} p(L_r = l_r)$, where l_r is the length of path r.

Our protocol, runs by making each node collect in a table all the source routes that it is part of – recall that since the network is assumed to be bidirectional, a node can extract the route request ID, the initiator and the target from the route request packet, save them in a temporary table, and then, if a route reply packet carrying a source route with the same initiator and target is observed within a pre-determined time interval, the node can associate the source route with the route request ID, and save both in a long-term table.

This mechanism brings about our security model. Since the common randomness established between two nodes by our algorithm consists of the source routes, it should be clear that several other nodes can be privy to this information. For instance, all the nodes included in a particular source route have full knowledge of this route. Moreover, it is likely that the route reply packet carrying a source route can be overheard by malicious eavesdroppers that are not part of the source route at all. Therefore, to achieve a level of security, two nodes will have to gather a large collection of source routes, such that *none of the other nodes that appear in any of the source routes in this collection has access to all the routes in the collection*. Unfortunately this is not enough, because it is still possible that one of the nodes, most likely a node that is part of many – though not all – routes in the collection, eavesdropped on all the remaining routes that it is not part of.

We deal with this problem by making an additional assumption: we assume that any two source routes are exchanged under independent and uniformly distributed network arrangements. That is, for the exchange (route discovery) of each source route, all the nodes in the network are distributed uniformly, and independently of other exchanges, in their predefined area. Moreover, the network remains the same for the entire duration of the route discovery and the associated data transmission. These assumptions are appropriate when the network nodes move around fast relative to the time between two different route discovery phases, but slow relative to the duration of a single communication session. This means that for any source route, the probability that any node which is *not* itself part of the route overhears the route (by overhearing a route reply or a data packet) is only a function of the network parameters. In the remainder of this section, we show how to compute the probability that an eavesdropper Eve knows a source route of which it is not part.

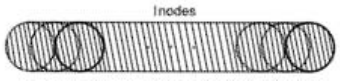


Fig. 2: The area covered by l nodes

Denote the binary random variable encoding whether an eavesdropper Eve overhears a source route r by $K_{Eve}(r)$. Then $p(K_{Eve}(r) = 1)$ depends on: (a) Eve's reception radius, (b) the total area of the network (all the places where Eve could be during the communication session corresponding to source route r), and (c) the length of the path. The computation is described in figure 2, where it can be observed that the worst-case scenario for a path of length l is when all the l nodes are arranged in a straight line. In this case, we can use the following worst-case approximation (obtained by first calculating the area of a circular segment):

$$p(K_{Eve}(r) = 1|L_r = l) = \\ = \frac{l\pi d_e^2 - 2(l-1)d_e^2(\frac{\pi}{3} - \frac{\sqrt{3}}{4})}{S_{total}} = \frac{d_e^2(1.91 \cdot l + 1.23)}{S_{total}}, \quad (1)$$

where d_e is the maximum eavesdropping range (the radius of the circles in figure 2), which is assumed the same for each of the nodes (all nodes transmit with the same power, using isotropic antennas), and S_{total} is the total area of the predefined location where the nodes can move.

Finally, two additional assumptions are made: the attackers are purely passive eavesdroppers (as attackers – otherwise, they are allowed to initiate well-behaved communication, just like any other node), and they do not collude.

IV. Proposed Algorithm

In this section we introduce KERMAN, a *K*ey-*E*stablishment algorithm based on *R*andomness harvested from the source routes in a *MAN*ET employing the DSR algorithm. To establish secret common randomness between two nodes in the MANET, KERMAN uses the standard sequence of three steps outlined in Section I: advantage distillation, information reconciliation and privacy amplification.

A. Advantage Distillation

To accomplish advantage distillation, every node in the network has to maintain a new table called the *Selected Route Table*, or SRT. The SRT contains source routes that include that node's address. To demonstrate how the SRT is built, we consider the following example. Take the scenario in figure 1, in which node 1 and 6 are the source and the destination,

[1]It should be noted that our algorithm should work (albeit with some reduction in performance) even if the network is not bidirectional. In this case, the route request ID needs to be inserted in the route reply packet. The reduction in performance for this scenario follows from the security considerations – namely, more nodes are involved in the routing mechanism, and hence have access to the source route.

respectively. Since node 1 does not have any route to node 6, it generates and broadcasts a route request packet. Assume that the id of this packet is 14, which means that this is the fourteenth attempt that node 1 makes to reach node 6. As seen in figure 1, node 5 will generate the route reply from its own route cache (because we assumed that node 5 already knows how to reach node 6). The transmission path of the route reply from node 5 to node 1 has been illustrated in figure 1, and is consistent with a bidirectional network. Each intermediate node that receives this route reply inserts the source route in their own SRT. The SRT has three columns dubbed *RID*, *partial route* and *full route* respectively. RID is a tuple that consists *(Source IP, Destination IP, route request ID, route-reply-sender IP)*. In our scenario, nodes 1, 2, 3, 4 and 5 will all record an entry in their respective SRTs, with the RID 1-6-14-5. The intermediate nodes (2, 3 and 4) can obtain the route request ID by searching their own *route request tables* as discussed in Section II. The *partial route* field of the SRT entry identifies those other nodes that are supposed to have this particular route in their SRT – in this case, nodes 1, 2, 3, 4 and 5. The *full route* field is the entire route from source to destination, which will be used for data transmission (1,2,3,4,5,6 in this case). The SRTs of the nodes 1, 2, 3, 4 and 5 have the same following entry:

RID	Partial Route	Full Route
1-6-14-5	1-2-3-4-5	1-2-3-4-5-6

It should be noted that, because node 6 did not directly hear the route request from node 1, it has no way of determining the route request ID in the RID, and this is why it cannot store this entry in its SRT, although it will most likely learn the source route from the received data packets that follow the route discovery phase. Regardless, when discussing the security of the established secret common randomness, node 6 will be assumed to have full knowledge of the *full route*.

Each full route in a nodes' SRT is only available to a limited number of nodes in the network, i.e., those nodes which are included in in the source (full) route, along with some nodes who are not part of the source route but happen to overhear the route request and route reply exchange. The following proposition, the proof of which is available in [18], states that SRT entries are unique in the whole network.

Proposition 1: If two nodes have the same RID in their own SRTs, then the full routes associated with this RID in two SRTs are exactly the same.

B. Information Reconciliation

Let us assume that two nodes –call them Alice and Bob for simplicity – realize that they share a large number of routes in their SRTs. For instance, Alice could first notice that Bob is part of a large number of partial routes in her SRT, and could ask Bob to perform information reconciliation, with the purpose of eventually generating a shared secret key. Upon Bob's acceptance, Alice sends him the list of RIDs corresponding to the partial routes in Alice's SRT that include the address of Bob. Bob can then verify whether he already has the received RIDs in his SRT, and can send back to Alice only those RIDs that he could not locate. The information reconciliation is now complete. Alice and Bob share a set of full routes, which constitute their common randomness.

As mentioned in Section IV-A, the RIDs consist of the tuples (Source IP, Destination IP, route request ID, route-reply-sender IP) corresponding to each route request/ route reply pair. Moreover, it is possible that Alice and Bob are neither the source nor the destination, nor the route-reply sender. Thus, transmitting an RID in the clear, over a public channel, may expose up to five nodes of the route (source, destination, route-reply sender, Alice and Bob) to an eavesdropping adversary. While this does not prevent KERMAN from working, many of the full routes of length less than six would become useless for our purposes. Practical solutions can be employed to limit the amount of information that the reconciliation leaks to eavesdroppers – a good starting point is provided in [20]. However, for the purposes of this paper, we only consider the ideal case in which, when transmitting the RIDs from Alice to Bob, no information leaks about the contents of the RIDs, except the addresses of Alice and Bob (we assume that every node in the network can see that Alice an Bob exchange RIDs). This can be seen as an upper bound on the performance of KERMAN. A lower bound (when the RIDs are sent in the clear) is provided in [18].

C. Privacy Amplification

For the purposes of this section we shall represent the full routes as sets of node identifiers, or addresses. Alice and Bob share a list of common full routes. Now Alice and Bob can construct the set $\mathcal{M} = \{m_1, m_2, \ldots, m_h\}$ where m_i (we'll call it a *trimmed route*) is produced from the full route r_i, by removing the addresses of Alice and Bob. At this point, full routes and trimmed routes are in a one-to-one correspondence.

In the next step, Alice partitions the set of trimmed routes $\mathcal{M}$ into several *disjoint* subsets $\mathcal{M}_k \subset \mathcal{M}$ of various sizes h_k, such that, for any $\mathcal{M}_k = \{m_{1,k}, m_{2,k}, \ldots, m_{h_k,k}\}$, the probability that any node in the network has knowledge of all the h_k trimmed routes is less than a small security parameter ϵ_1. This means that, with probability larger than $1 - \epsilon_1$, there exists at least one trimmed route in $\mathcal{H}$ that Eve knows nothing about – note that this is true for any identity that Eve may take (except, of course Eve cannot be Alice or Bob). It is the full route corresponding to this trimmed route (different from any node's perspective) that constitutes the randomness of the generated secret.

To extract a secret from each of the sets $\mathcal{M}_k$, Alice first represents all the *full routes* by pre-determined binary strings of the same length. The length of the strings is determined as the logarithm to base two of the total number of possible full routes, in a practical scenario. For example, from our simulations, we noticed that full routes are limited to 15 nodes, which means that trimmed routes are limited to 13 nodes. In a network of 50 nodes, there are thus $\binom{48}{1}3! + \binom{48}{2}4! + \ldots + \binom{48}{13}15!$ possible full routes involving Alice and Bob, where the factorial terms account for all the possible arrangements. For example, there are $\binom{48}{1}$ trimmed routes of length 1, and their corresponding full routes have length 3 (this includes the unknown node that defines the trimmed route, Alice and Bob), and there are $3! = 6$ possible arrangements of these three nodes. This total number of possible full routes amounts to representing each full route on 78 bits. The binary sequences representing the *full routes* corresponding to the trimmed routes in $\mathcal{M}_k$ are then XORed together. The result is then inserted into a (k, ϵ_2)-randomness extractor [21], which

outputs a shorter bit string s_k – the secret. The secret s_k should satisfy the (ϵ_1, ϵ_2)-security defined below.

Definition 1: Smooth Minimum Entropy: From [16]: Let X be a random variable with alphabet $\mathcal{X}$ and probability distribution of $P_X(x)$, and let $\epsilon_3 > 0$. The ϵ_3-smooth min-entropy of X is defined as $H_\infty^{\epsilon_3}(X) = -log \max_{Q_X \in \mathcal{B}^{\epsilon_3}(P_X)} Q_X(x)$, where the maximum ranges over the ϵ_3-ball $\mathcal{B}^{\epsilon_3}(P_X)$ [16]. When $\epsilon_3 = 0$, smooth min entropy becomes the min entropy.

Definition 2: In the context of a MANET, a piece of secret common randomness s_k established between two nodes Alice and Bob is called (ϵ_1, ϵ_2)-secure if, with probability larger than $1-\epsilon_1$, the secret s_k is ϵ_2-close to uniform from the perspective of any node in the network, except Alice and Bob.

It has been shown in [16] that the number of completely random bits that can be extracted from a bit sequence should be upper bounded by, but very close to, the (smooth) min-entropy of the sequence. Thus, in this paper, we shall only focus on the (smooth) minimum entropy of a full route, viewed from the perspective of an eavesdropper who does not know anything about the associated trimmed route. This minimum entropy is a good indication of the number of secret random bits that can be extracted from each set $\mathcal{M}_k$. To calculate it, we need to compute a probability distribution that characterizes Eve's belief about the unknown full route. The task is not straightforward. First routes of different lengths have different probabilities of appearing in SRTs (these depend on the network parameters). Second, if a route is longer, then the probability that Eve has accidentally overheard it is larger – recall (1). Therefore, we start off with an empirically-derived prior $p(L_r = l_r)$ denoting the probability that the unknown route has length l_r, and with $p(K_{Eve}(r) = 0|L_r = l_r) = 1 - p(K_{Eve}(r) = 1|L_r = l_r)$ from (1), and we compute

$$p(L_r = l_r|K_{Eve}(r) = 0) = \\ = \frac{p(L_r = l_r)p(K_{Eve}(r) = 0|L_r = l_r)}{\sum_l p(L_r = l)p(K_{Eve}(r) = 0|L_r = l)}. \quad (2)$$

As explained above, from Eve's perspective there are $\binom{N-2}{l_r-2} l_r!$ equally-likely full routes of length l_r and containing Alice and Bob, where N is the total number of nodes in the MANET. Thus we can write the probability that the unknown full route is r (where r has the length of l_r) as:

$$p(r|K_{Eve}(r) = 0)) = \frac{p(L_r = l_r|K_{Eve}(r) = 0)}{\binom{N-2}{l_r-2} l_r!}, \quad (3)$$

1) The partitioning algorithm: The remaining question is how many subsets $\mathcal{M}_k$ we can form. To solve this problem, for any pair of nodes we organize the full set of all trimmed routes $\mathcal{M}$ as a *selection matrix*. In the *selection matrix*, a row corresponds to one of the trimmed routes in $\mathcal{M}$. A column corresponds to a node's address. There are 48 columns (one for each node in the MANET, except Alice and Bob). Each entry in the matrix is the probability that the node in the respective column knows the full route corresponding to the respective row. The selection matrix can be represented as follows:

$$\begin{array}{c} \\ m_1 \\ m_2 \\ \vdots \\ m_h \end{array} \begin{array}{c} \begin{array}{cccc} node\ 1 & node\ 2 & \dots & node\ t \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1t} \\ a_{21} & a_{22} & \dots & a_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nt} \end{pmatrix} \end{array}$$

where a_{ij} is the probability that node j knows full route i. For example, when node j is a part of the full route corresponding to the trimmed route i, then $a_{ij} = 1$. Otherwise, $a_{ij} = p(K_j(i) = 1|L_i = l_i)$, where l_r is the length of route i. The partitioning algorithm consists of constructing *distinct* sub-matrices $\mathcal{M}_k$, each consisting of h_k rows of $\mathcal{M}$, such that the product of the entries in each column of $\mathcal{M}_k$ be less than ϵ_1. We shall informally call this property ϵ_1-security, and we shall use the terms *subset* and *sub-matrix* interchangeably. An optimal partition maximizes the number of sub-matrices $\mathcal{M}_k$ with the ϵ_1-security property.

The naïve algorithm proposed in [18] takes a greedy approach: it builds $\mathcal{M}_1$ by selecting the first row in the selection matrix, and adding the next row in the selection matrix, until the column-wise product condition holds. Then it moves to the next row, and starts building $\mathcal{M}_2$, and so on, until we run out of rows in $\mathcal{M}$.

As an alternative to the naïve algorithm, we provide a better-performing (but more complex) heuristic algorithm, that goes as follows. Starting with the original selection matrix, we inspect all sub-matrices of two rows, and check whether any of them satisfies the ϵ_1-security property. If any such disjoint sub-matrices are found, we count the corresponding subsets of rows $\mathcal{M}_k$, we update the selection matrix by removing these rows from the original selection matrix, and we go on to inspect all the sub-matrices consisting of three rows of the updated selection matrix.

So far, the algorithm seems to perform optimally. However, the main problem that prevents the algorithm from being optimal arises because in general several partially-overlapping sub-matrices can be formed at each step. For example, consider a scenario where two sub-matrices of two rows have been found to satisfy ϵ_1-security: say these sub-matrices are the one consisting of rows 2 and 6 of the selection matrix, and the one consisting of rows 2 and 8. Clearly, only one of them can be considered for privacy amplification, lest we compromise the entropy of the secret key. We now have to decide which of the two choices results in an updated selection matrix that is more likely to perform better in future partitions. For our example, if row 6 is less than row 8 (i.e. component i or row 6 is less than component i of row 8, for all i), then we should select the sub-matrix containing rows 2 and 8, because row 6 might prove more useful in the future. But because such an ordering of matrices is usually not clear-cut, we proceed to define our own *partial order*, which is essentially sub-optimal, and responsible for the sub-optimality of our heuristic algorithm.

Definition 3: Average-Column-Product Sub-Optimal Partial Order (ACP-PO): For any two partially-overlapping sub-matrices $\mathcal{M}_i$ and $\mathcal{M}_j$ of the selection matrix, with $\mathcal{M}_i \bigcap \mathcal{M}_j = \mathcal{M}_{ij} \neq \emptyset$, we say that $\mathcal{M}_i$ is better than $\mathcal{M}_j$ in the ACP-PO sense, and write $\mathcal{M}_i \prec \mathcal{M}_j$ if the mean of the column-wise products of elements of $\mathcal{M}_i$ is less than mean of the column-wise products of elements of $\mathcal{M}_j$. We say that $\mathcal{M}_i$ is at least as good as $\mathcal{M}_j$ in the ACP-PO sense, and write $\mathcal{M}_i \preceq \mathcal{M}_j$ if the mean of the column-wise products of elements of $\mathcal{M}_i$ is less than or equal to the mean of the column-wise products of elements of $\mathcal{M}_j$.

Our algorithm is illustrated by the pseudo-code fragment of Algorithm 1. The algorithm starts by checking whether at least one sub-matrix verifying the ϵ_1-security condition can be found – that is, whether the whole original selection matrix satisfies

ϵ_1-security. The algorithm then finds all the sub-matrices of $SubsetSize$ rows of M that satisfy ϵ_1-security, and orders them according to the ACP-PO defined above. Recall that this partial order is only meaningful for two sub-matrices that have at least one row in common, but our algorithm orders the whole list of subsets anyway. After sorting all sub-matrices in descending ACP-PO we pick and process the first sub-matrix. We then make sure that the rows we already picked are not going to be considered again, by updating the ordered list and the selection matrix M. The algorithm will then continue to select sub-matrices from the remaining list, and when the list becomes empty, it switches the search to sub-matrices with more rows.

Algorithm 1 Heuristic Algorithm

```
M=Selection Matrix;
SubsetSize = 2;
NumberSubsets = 0;
L =Number of rows in M;
while Subsetsize ≤ L AND M satisfies ε1-security do
    for All combinations M_k of Subsetsize rows do
        if M_k satisfies ε1-security then
            Calculate average of column-wise products;
    Sort subsets based on the average of column-wise products;
    while Not End of List do
        Select and process first subset in the ordered list and increment NumberSubsets;
        Delete from the list all subsets that share rows with the selected subset;
        Update M by deleting all the rows in the selected subset;
    Increment SubsetSize;
    Update L;
```

To gain more insight into the algorithm's complexity, consider a case in which the initial selection matrix has h_1 rows. In the first stage, the algorithm examines $\binom{h_1}{2}$ partitions (sub-matrices), and if it finds any that satisfy ϵ_1 security, it updates the selection matrix, which will end up with $h_2 \leq h_1$ rows. The second stage inspects $\binom{h_2}{3}$ partitions, and so forth. All in all the heuristic algorithm should examine $\binom{h_1}{2} + \binom{h_2}{3} + \binom{h_3}{4} + ...$ partitions, which for most cases should be a lot less than 2^{h_1}. However, if there is no reduction in the number of rows in the first stages, the algorithm has to explore all 2^{h_1} partitions. Several simplifying solutions can be considered to avoid this situation: (1) if it is observed that over a pre-determined period of time the algorithm produces only sub-matrices with at least S_0 rows, then the algorithm can start with $SubsetSize = S_0$ rather than $SubsetSize = 2$; (2) the algorithm can test whether at least two sub-matrices are even possible, by testing whether the whole (updated) selection matrix M satisfies ϵ_1^2-security. If it does not, then the algorithm can stop searching for sub-matrices, and can process the whole selection matrix as a single sub-matrix.

V. Simulation Results

A. OPNET Simulation and Results

The proposed protocol has been simulated in OPNET Modeler, using the parameters indicated in table I. This choice of parameters results in a maximum eavesdropping range of $d_e = 12$m.

Each node sends packets to four random destinations. The number of full routes vs the full route length is shown in figure 3, and the empirically-derived prior $p(L_r = l_r)$ looks similar. To calculate the min entropy, we need the probability distribution of the unknown full route. According to (3), this is $p(r|K_{Eve}(r) = 0)) = \frac{p(L_r = l_r | K_{Eve}(r)=0)}{\binom{48}{l_r - 2} l_r!}$. Due to the obtained values being more than 20 orders of magnitude apart, we chose to represent $p(r|K_{Eve}(r) = 0))$ in table II. It can be easily seen that $H_{min}(r|K_{Eve}(r) = 0)) = -\log_2(0.00062) = 10.66$. There are $\binom{48}{1}3! = 288$ full routes of length 3, which implies that smoothing out the probability distribution $p(r|K_{Eve}(r) = 0))$ with an acceptably-low security parameter [16] has no noticeable effect on the value of the min entropy. Thus, about 10 bits can be extracted from each subset $\mathcal{M}_k$ of full routes. In figure 4 we show the number of pairs of nodes that share selection matrices with a given number of rows. Clearly, the larger the number of rows in the shared selection matrix, the higher the potential for generating more shared secret bits.

TABLE I: Simulation Parameters

Simulation Parameters	Value
Network Size	100m*100m
Number of Nodes	50
Simulation Duration	600(sec)
Transmit Power(w)	.005
Packet Reception-Power Threshold(dBm)	-55
Speed(meters/seconds)	uniform(.5,1)
Packet Inter-Arrival Time(seconds)	exponential(1)

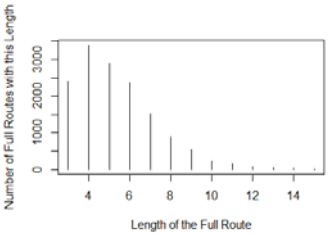


Fig. 3: Number Of Full Routes vs. Full Route Length

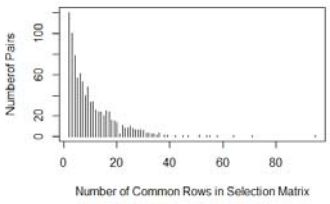


Fig. 4: Number Of Pairs vs. Number of Rows in their shared Selection matrix

For $\epsilon_1 = 10^{-3}$, the number of sub-matrices satisfying ϵ_1-security, produced by the heuristic algorithm for the whole network, is shown in table III.

We also calculate the maximum achievable total network-wide number of shared random bits (between all the possible pairs in the network), B_{total} – which is $4.98 \cdot 10^3$ bits for naïve algorithm and $5.08 \cdot 10^3$ bits for heuristic algorithm [2]. If we compare the result of the heuristic algorithm with that of the naïve algorithm, we can see an increase in the total number of bits from $4.98 \cdot 10^3$ to $5.084 \cdot 10^3$. Additionally, the numbers of subsets with a given size (number of rows) are shown for the whole network in figure 5 and figure 6, for the naïve algorithm and the heuristic algorithm, respectively. It can be clearly seen that the heuristic algorithm produced a lot more subsets of smaller size, which implies that it produces more subsets overall.

[2]For example, for $\epsilon_1 = 10^{-3}$, the heuristic algorithm produces $B_{total} = 10.66 \cdot (171 \cdot 1 + 70 \cdot 2 + 39 \cdot 3 + 25 \cdot 4 + 9 \cdot 5 + 2 \cdot 6 + 1 \cdot 7 + 2 \cdot 8 + 1 \cdot 9)$.

TABLE III: Number of subsets, obtained by the naïve and heuristic algorithms with $\epsilon_1 = .001$, when considering all full routes of length at least 3.

No. of Subsets	1	2	3	4	5	6	7	8	9
No. of Pairs-naïve	215	75	22	6	1	0	1	0	0
No. of Pairs-heuristic	171	70	39	25	9	2	1	2	1

TABLE II: Probability Distribution of an Unknown Full Route, from Eve's Perspective

Length Of the Full Route	3	4	5	6	7	8	9	10	11	12	13	14	15
probability	0.00062	9.02E-06	9.7E-08	1.1E-09	1.1E-11	1.1E-13	1.2E-15	9.9E-18	1E-19	1E-21	1.6E-23	2E-25	2E-27

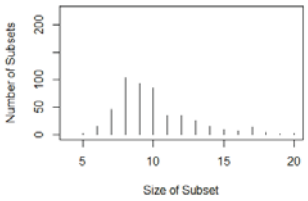


Fig. 5: Number Of subsets of a given size (number of rows), vs. Subset size, for the naïve algorithm – network-wide results.

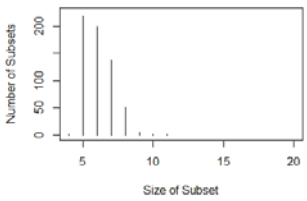


Fig. 6: Number Of subsets of a given size (number of rows), vs. Subset size, for the heuristic algorithm – network-wide results.

TABLE IV: Number of subsets, obtained by the naïve and heuristic algorithms, when considering only full routes of length at least 4.

No. of Subsets	1	2	3	4	5	6	7	8
No. of Pairs-naïve	195	60	11	2	2	0	0	0
No. of Pairs-heuristic	165	56	31	11	5	1	0	1

B. Increasing The Secret's Length by Spoiling Knowledge

Spoiling knowledge was introduced in [19] as a means of (publicly) adjusting a probability distribution to increase its min entropy. In our specific example, this translates to purposely discarding the most likely full routes from the SRT. But since all routes of the same length have the same probability (from Eve's perspective), we can only increase the min entropy by discarding all the routes of a given length. The downside, of course, is that the number of partitions satisfying the properties outlined in Section V-A also decreases.

For our specific scenario, disregarding the full routes of length 3 yields a min entropy of roughly $H_{min}(r|K_{Eve}(r) = 0)) = -\log_2(9.02 \cdot 10^{-6}) = 16.76$ bits. The number of subsets produced for the whole network, for $\epsilon_1 = 10^{-3}$, is shown in table IV, for our two different partitioning algorithms. In this case, B_{total} is $6.13 \cdot 10^3$ bits for naïve algorithm and $7.59 \cdot 10^3$ bits for heuristic algorithm. It is interesting to note that the spoiling knowledge technique achieves a gain of roughly 23% and 49%, in the naïve algorithm and in the heuristic algorithm respectively.

VI. Conclusions and Future Work

We have shown that the randomness inherent in an ad-hoc network can be harvested and used for establishing shared secret keys, and we provided an algorithm for doing so. Our algorithm can employ either of two partitioning algorithms, to trade complexity for an increase in secret lengths. For practical network parameters, we have demonstrated that after only ten minutes of use, thousands of shared secret bits can be established between various pairs of nodes. This number can be further increased by the spoiling knowledge technique of [19]. While we showed how this works at the entire-network level, a better option might be to let each one of the pairs of nodes decide whether using the spoiling knowledge technique is advantageous or not.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[2] S. K. Park and K. W. Miller, "Random number generators: good ones are hard to find," *Communications of the ACM*, vol. 31, no. 10, pp. 1192–1201, 1988.

[3] B. Sunar, "True random number generators for cryptography," in *Cryptographic Engineering*. Springer, 2009, pp. 55–73.

[4] U. E. Maurer, "Secret key agreement by public discussion from common information," *Information Theory, IEEE Transactions on*, vol. 39, pp. 733–742, May. 1993.

[5] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography – Part I: secret sharing," *Information Theory, IEEE Transactions on*, vol. 39, pp. 1121–1132, July 1993.

[6] ——, "Common randomness in information theory and cryptography– Part II: cr capacity," *Information Theory, IEEE Transactions on*, vol. 44, no. 1, pp. 225 –240, jan 1998.

[7] J. W. Wallace and R. K. Sharma, "Automatic secret keys from reciprocal mimo wireless channels: measurement and analysis," *Trans. Info. For. Sec.*, vol. 5, pp. 381–392, September 2010.

[8] M. Bloch, J. Barros, M. Rodrigues, and S. McLaughlin, "Wireless information-theoretic security," *Information Theory, IEEE Transactions on*, vol. 54, no. 6, pp. 2515 –2534, june 2008.

[9] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 2, pp. 240 –254, june 2010.

[10] A. Agrawal, Z. Rezki, A. Khisti, and M. Alouini, "Noncoherent capacity of secret-key agreement with public discussion," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 3, pp. 565 –574, sept. 2011.

[11] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 1422 –1430.

[12] K. Ren, H. Su, and Q. Wang, "Secret key generation exploiting channel characteristics in wireless communications," *Wireless Communications, IEEE*, vol. 18, no. 4, pp. 6 –12, august 2011.

[13] T.-H. Chou, S. Draper, and A. Sayeed, "Key generation using external source excitation: Capacity, reliability, and secrecy exponent," *Information Theory, IEEE Transactions on*, vol. 58, no. 4, pp. 2455 –2474, April 2012.

[14] C. Ye and P. Narayan, "Secret key and private key constructions for simple multiterminal source models," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 639 –651, Feb. 2012.

[15] A. Khisti, S. Diggavi, and G. Wornell, "Secret-key generation using correlated sources and channels," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 652 –670, Feb. 2012.

[16] R. Renner and S. Wolf, "Simple and tight bounds for information reconciliation and privacy amplification," pp. 199–216, 2005.

[17] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Tech. Rep., 2007. [Online]. Available: http://tools.ietf.org/html/rfc4728

[18] M. R. K. Shoja, G. T. Amariucai, S. Wei, and J. Deng, "Secret common randomness from routing metadata in ad-hoc networks," *IEEE Transactions on Information Forensics and Security accepted for publication*, 2016.

[19] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *Information Theory, IEEE Transactions on*, vol. 41, no. 6, pp. 1915–1923, 1995.

[20] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion." Springer-Verlag, 1994, pp. 410–423.

[21] R. Shaltiel, "An introduction to randomness extractors," in *Automata, languages and programming*. Springer, 2011, pp. 21–41.

Performance Comparison of AES-CCM and AES-GCM Authenticated Encryption Modes

Levent Ertaul, Anup Mudan, Nausheen Sarfaraz
CSU East Bay, Hayward, CA, USA.
levent.ertaul@csueastbay.edu, amudan@horizon.cueastbay.edu, nsarfaraz@horizon.csueastbay.edu

Abstract — **In data security, especially in mobile devices, it has long been understood that to meet the highest compliance standards, Authenticated Encryption is required. Encryption alone is not enough to provide the utmost level of security in various mobile applications. This paper proposes the implementation of Authenticated Encryption Mode, CCM in our application. This work also shows a comparison of the performance analysis of AES-CCM and AES-GCM modes. The choice to use Android Java programming language was made in order to create an Android application which sends and receives text messages between two parties by sharing a secret key, and uses an authentication feature. The execution of the algorithm is performed in Android Studio and the implementation of the code is accomplished using Android API.**

I. Introduction

Authenticated Encryption is a process of ensuring that both ends of a connection are completely secure. Mobile operating systems in today's world are vulnerable when it comes to hackers. Eighty percent of the world's cellphones use an open source operating system such as Android [8]. Open source allows third parties to change the original code according to their own requirements and needs. This can often leave open loop holes and back doors that hackers will try to exploit. For this, it is not sufficient to use basic encryption techniques to protect information. Authenticated Encryption, or AE, addresses these issues by creating a more secure and bulletproof connection. AE security protects the user and service being used from the inherent flaws in open source software, and ensures that information within a session is not being compromised. In addition to providing authentication and confidentiality, AE provides a strong protection from various attacks like replay attacks, chosen cipher-text attacks, and the man-in-the-middle attack.

Over the last decade, there has been significant amount of research and effort involved to invent the dedicated AE modes CCM, GCM, EAX, and OCB [7]. Rather than using the authenticity and privacy techniques separately, these AE schemes provide more proficient results and have very few chances of being incorrect. In order to ensure safety of the information, it was suggested to combine authentication mechanism such as MAC with the encryption algorithms. The combinations were applied in multiple secure and insecure ways [6]. This paper talks about the comparison of the CCM (CTR + CBC-MAC) mode and the GCM (Galois Counter Mode) [9] mode of operation, which are symmetric key block cipher algorithms defined and used in security systems. The key features like authenticity, integrity and confidentiality are achieved by these modes [18]. CCM is defined in IEEE 802.11i, IPsec [19], TLS 1.2 [20] and uses Advanced Encryption Standard [13] [5] as its cryptographic algorithm. AES is the standard recognized by National Institute of Standards and Technology (NIST) in 2001 [2] and specified in Federal Information Processing Standard (FIPS) [4].

GCM is used in various security standards such as the IEEE 802.1AE for frame data encryption in the Ethernet [15], the IEEE P1619.1 for encrypting hard disks [16], IEEE 802.11AD, and RFC 4106 IPsec [17]. It is based on a parallelization process which generates ciphertexts and an authentication tag simultaneously. It uses counter mode and a hash function over Galois Field (2 ^ 128) to generate a tag. It consists of Galois Field (GF) multiplier adders. In counter mode, the counter blocks are numbered in a sequential manner and the encryption function is performed on these blocks. The output of this function is XORed with the plaintexts to produce ciphertexts. A hash function is used to generate the tag by combining the ciphertext and an authentication code to check the integrity of the data [9].

This paper is organized into the following sections: Section II proposes the approach we followed using the AES-CCM algorithm. Section III discusses the AES-CCM algorithm. Section IV represents the implementation of this algorithm using Android API. Section V shows the performance analysis of CCM and GCM mode and the comparison results. Section VI proposes the conclusion of this paper.

II. Traditional Encryption v/s AES-CCM

Companies sometimes prefer traditional encryption methods when it comes to using mobile applications. The main focus of the traditional encryption schemes is to provide confidentiality, but they do not protect from malicious tampering or data being modified intentionally by the attackers. These methods do not provide the level of security required, and in fact they can be vulnerable to an informed hacker. Authenticated encryption schemes are the alternative methods. Using AE schemes, many different approaches are taken into consideration, i.e. Encrypt-then-MAC, Encrypt-and-MAC and MAC-then-Encrypt [14].

The solution proposed in this paper uses MAC-then-encrypt scheme in which the MAC value is generated first, and then the data and MAC are encrypted using counter mode. This would make it hard for the attacker to obtain the MAC value in order to perform attacks. The following figure 2.1 depicts the approach being implemented.

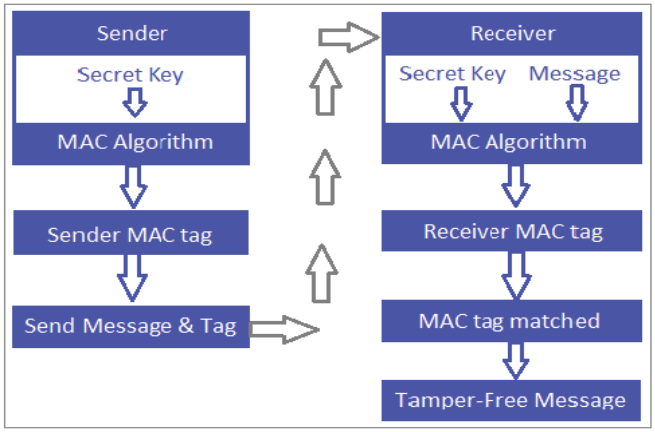


Figure 2.1 MAC-then-encrypt mechanism

III. The AES-CCM Algorithm

Advanced Encryption Standard, or AES, [13] is the standard known for a symmetric block cipher mechanism that uses 128 bits, 192 bits and 256 bits of key sizes. CCM is an Authenticated Encryption Standard which is based on a key management structure. In this algorithm, the plaintext is divided into block ciphers of 128 bits size. The modes of operations used in AES-CCM are counter mode (CTR) with Cipher Block Chaining and Message Authentication Code (CBC-MAC). They perform generation-encryption and decryption-verification functions [3]. The confidentiality feature is achieved in CTR mode by AES and the authentication is achieved in CBC-MAC with the MAC value generated.

In AES-CBC-MAC, the encryption function is applied to the first block to generate a cipher. Then the cipher result is XORed with the second block to obtain the next result. The process keeps going on for all the remaining blocks until the final value MAC is obtained, which is used in CTR mode encryption. The following 3.1 shows the block diagram of AES-CBC-MAC.

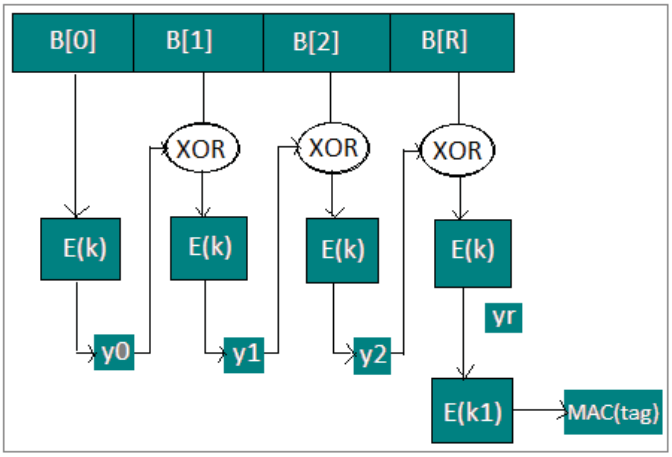


Figure 3.1 Block diagram of AES-CBC-MAC

In AES-CTR, different cipher blocks are produced which are dependent on nonce value. The CTR mode is applied to MAC and the payload to obtain the cipher-text [1]. CCM is not compatible with steam ciphers and does not work with the Data Encryption Standard which supports a 64 bits of block size. It works in the packet environment where all of the data is available in storage beforehand [3]. The following figure 3.2 shows the block diagram of AES-CTR mode.

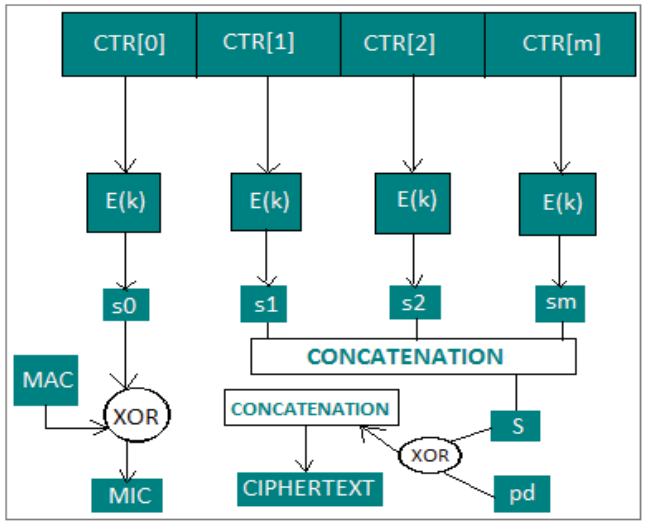


Figure 3.2 Block diagram of AES-CTR

The input elements of CCM are: the valid payload (*pd* < *2^64*) (data which is authenticated and encrypted), the valid nonce (*nc* < *2^61*) (must be unique), and the valid associated data (*ad* ≤ *256 bits*) (which is authenticated but not encrypted). The nonce is applied to the payload and the associated data. The secret key (*k*) to the block cipher is generated uniformly at random whose size is 128 bits. CCM only works with the forward cipher function [3].

A. Generation-Encryption

In generation-encryption mechanism, cipher block chaining is applied to the payload (*pd*), the nonce (*nc*), and the associated data (*ad*), to generate MAC. The MAC length (*Mlen*) is always greater than or equal to 64 bits. Then the counter mode encryption is applied to the MAC and payload to convert it into cipher-text [3].

Prerequisites:
The various prerequisites that are required are as follows: the cipher block algorithm, key *k,* counter generation function, formatting function, MAC length *Mlen.*

Input:
The input values required are: valid payload *pd* of length *pdlen* bits; valid associated data *ad;* valid nonce *nc.*

Output:
The output will be cipher-text *C.*

Steps:

1. Apply the formatting function to *(nc, ad, pd)* to produce the blocks *B0, B1,....., Br*
2. Set *Y0 = CIPH k(B0)*
3. For *I = 1 to r,* do *Yi = CIPH k(Bi XOR Yi-1)*
4. Set *MAC = MSBMlen (Yr)*
5. Apply the counter generation function to generate the counter blocks *CTR0, CTR1, ..., CTRm,* where *m = pdlen/128*

6. For *j = 0 to m,* do *Sj = CIPH k(CTRj)*
7. Set *S = S1 || S2 || .... || Sm*
8. Return *C = (pd XOR MSB pdlen(S)) || (MAC XOR MSB Mlen(S0))*

B. Decryption-Verification

In decryption-verification mechanism, counter mode decryption is performed to get the MAC value and its corresponding payload. Cipher block chaining is applied to the payload, the nonce received, and the associated data received to check if the MAC is correct. If the verification succeeds that means that inputs are generated from the source and have access to the key [3]. MAC plays the most important role as it can keep away security threats and can protect data from being modified.

Prerequisites:
The various prerequisites that are required are as follows: Cipher block algorithm; Key *k;* Counter generation function; Formatting function; and Valid MAC length *Mlen.*
Input:
The main input values required are: associated data*, ad;* nonce*, nc;* ciphertext *C* of length *cplen* bits*.*

Output:
The output will be either payload *pd* or *INVALID*.

Steps:

1. If *cplen* ≤ *Mlen,* then return *INVALID*
2. Apply the counter generation function to generate the counter blocks *CTR0, CTR1, ..., CTRm*
3. For *j = 0 to m,* do *Sj = CIPH k(CTRj)*
4. Set *S = S1 || S2 || ..... || Sm*
5. Set *pd = MSB cplen – Mlen(C) XOR MSB cplen – Mlen(S)*
6. Set *MAC = LSBMlen (C) XOR MSB Mlen(S0)*
7. If *nc, ad or pd* is not valid*,* then return *INVALID,* else apply the formatting function to *(nc, ad, pd)* to produce the blocks *B0, B1, ..., Br*
8. Set *Y0 = CIPH k(B0)*
9. For *I = 1 to r,* do *Yj = CIPH(Bi XOR Yi-1)*
10. If *MAC ≠ MSBMlen(Yr),* then return *INVALID,* else return *pd*

IV. Implementation

The following tables I and II show the hardware and software specifications of the device we used.

A. Specification

Table I. Hardware Specification

Type	**Specification**
Device Type	Mac OS X
Processor	2.5 GHz Intel Core i7
RAM	16 GB
Operating System	Android version 5.1 Lollipop

Table II. Software Specification

Type	**Specification**
Android Programming Language	Java
Android Studio	Version 2.0
Android API	Spongy Castle
Android Virtual Machine	Genymotion

B. Screen Shots

We created and executed our CryptUtil application using Android Studio. We made two Android Virtual Machines for the sender and the receiver side, up and running. The screen shots are taken from the emulator.

Once the application is launched, it shows the main page for sender and receiver (Figure 4.1) which contains buttons to send and receive a message.

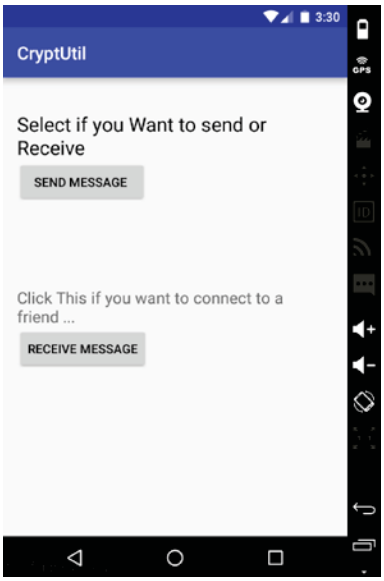


Figure 4.1 Main access page

Once the sender clicks on the Send Message button, the next page is displayed which tells the user to enter the IP address of the receiver side to get connected (Figure 4.2).

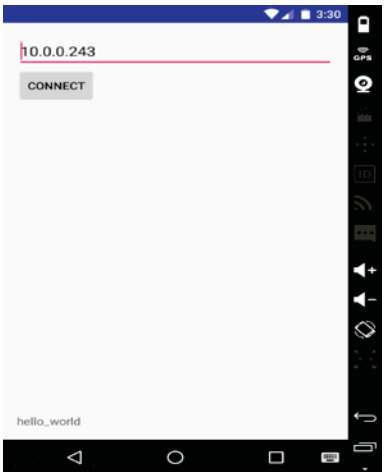


Figure 4.2 Entering IP address to connect to the Receiver

Once the sender enters the IP address, a page is displayed showing that the connection has been established (Figure 4.3).

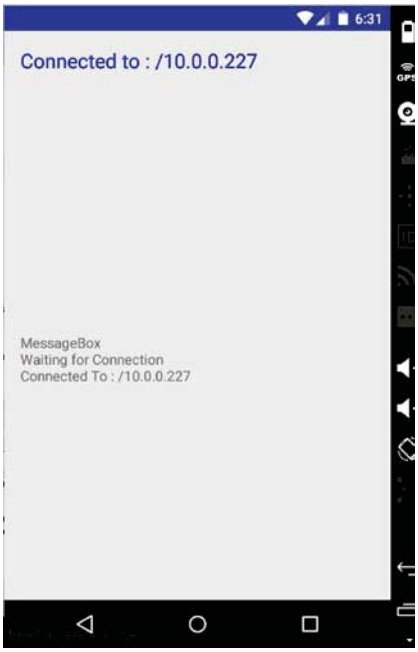


Figure 4.3 Connection established page

After the sender and receiver are connected, the page showing "Enter Your Message To Encrypt" field, and "Enter Your Password Key", field is displayed (Figure 4.4). This page also includes a text field to enter a AEAD value. The AEAD value has to match the default value already set in order to avoid an error.

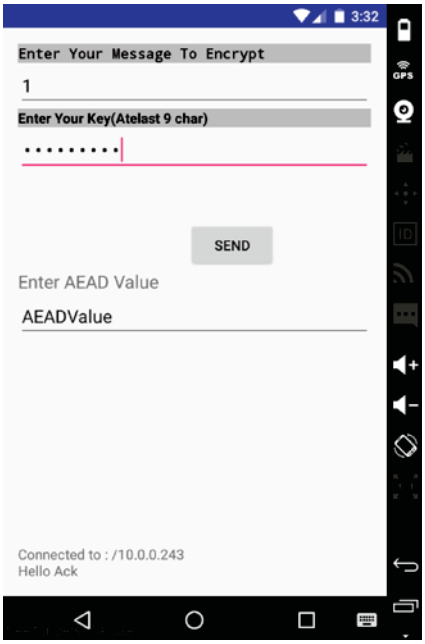


Figure 4.4 Enter message and key page on sender's side

Once the sender clicks on the Send button, the receiver receives the encrypted text i.e. cipher-text. Also, a dialog box appears for the receiver to enter the matching AEAD value and the password key in order to decrypt the text message (Figure 4.5).

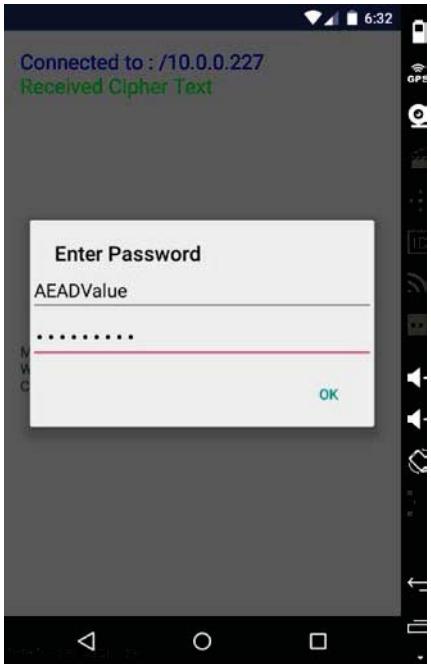


Figure 4.5 Enter password page on receiver's side

If the entered AEAD value and the password key both match the shared value and the key of the sender, then the decryption process is successful on the receiver's end (Figure 4.6). This page also shows the time it took to decrypt the text message in microseconds.

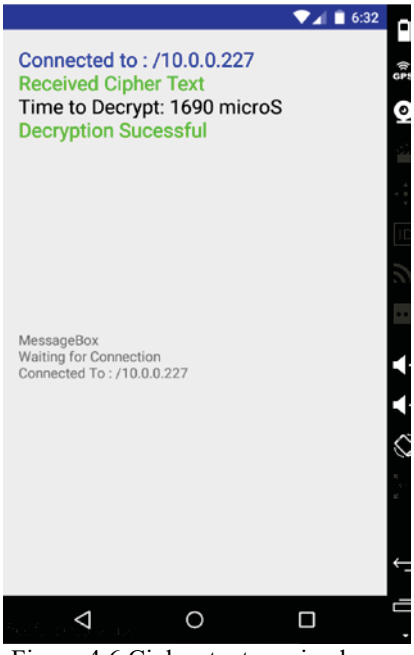


Figure 4.6 Cipher-text received page

If the password key or AEAD value entered do not match the shared key and the value, then the decryption process cannot be performed and it will end up displaying an error message (Figure 4.7).

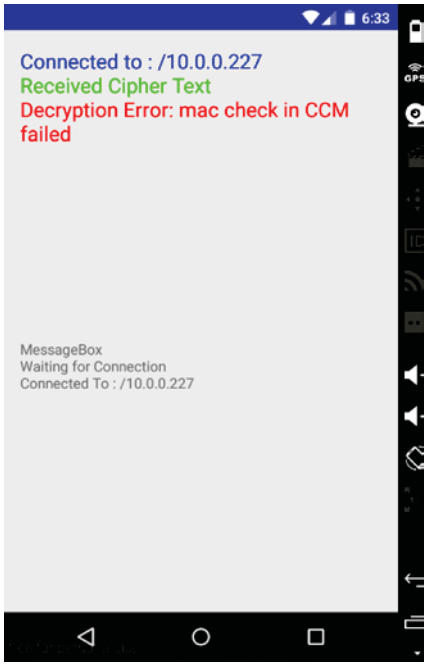


Figure 4.7 Decryption error page

V. Performance Analysis and Results

A. *Comparison using different AEAD values of 9 chars, 16 chars and 24 chars in AES-CCM*

In the following graphs (figures 5.1, 5.2 and 5.3), we can see that as we increase the number of characters in AEAD value, the encryption and decryption time increases.

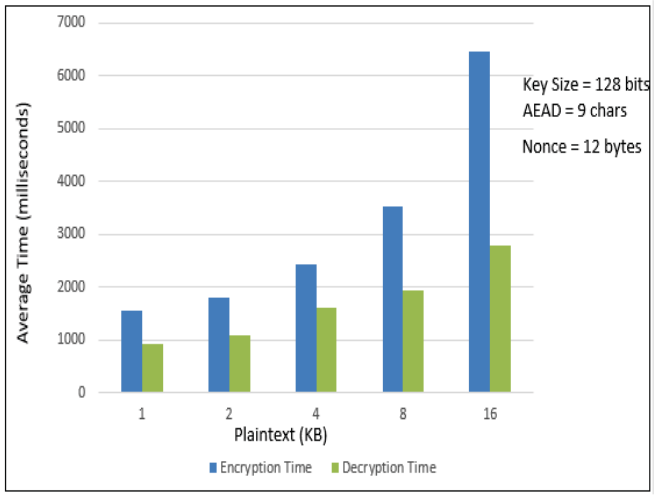


Figure 5.1 128 bits Key, 9 chars AEAD, 12 bytes Nonce

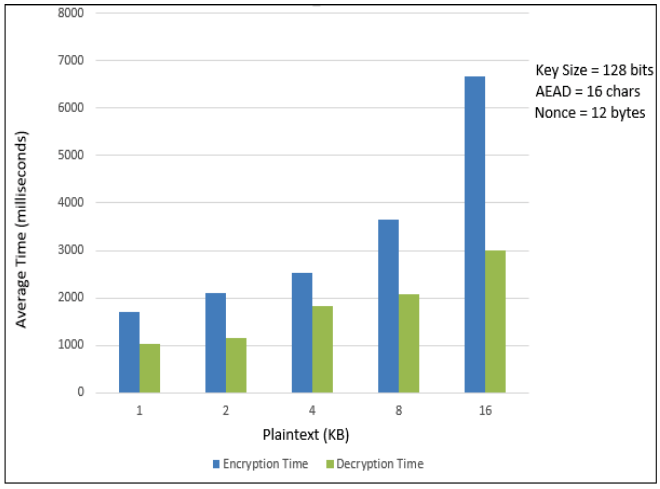


Figure 5.2 128 bits Key, 16 chars AEAD, 12 bytes Nonce

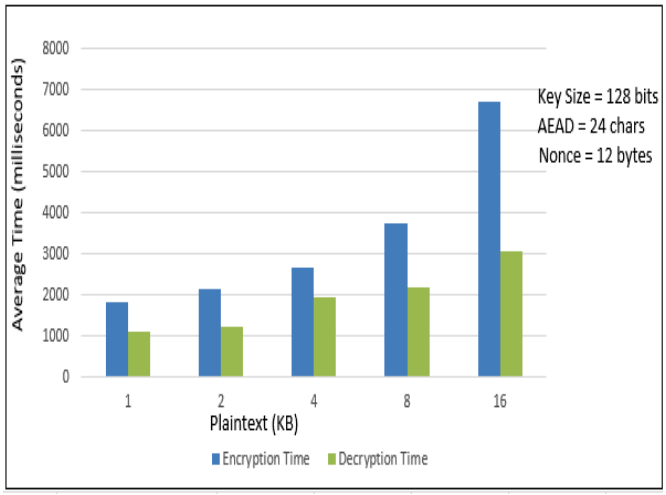


Figure 5.3 128 bits Key, 24 chars AEAD, 12 bytes Nonce

B. *Comparison using different key sizes of 128 bits, 192 bits and 256 bits in AES-CCM*

We can see the differences in the encryption time in the following figures with the variable key sizes. . As the key size increases, the encryption time rises. On the other hand, it shows a drop in the decryption time when the key size is changed from 128 bits to 192 bits.

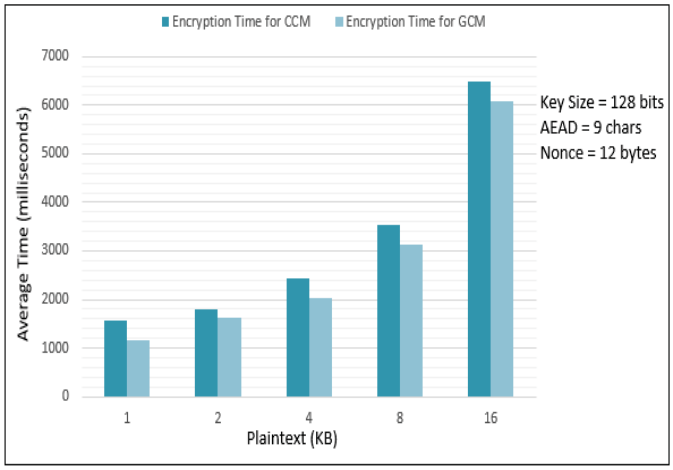


Figure 5.4 128 bits Key, 9 chars AEAD, 12 bytes Nonce

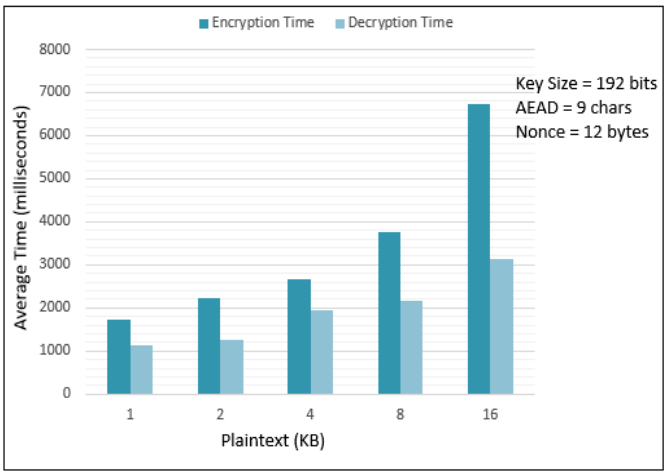


Figure 5.5 192 bits Key, 9 chars AEAD, 12 bytes Nonce

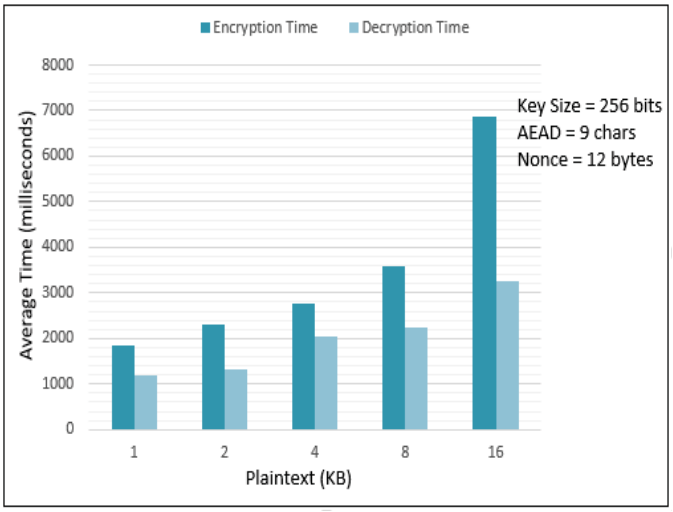


Figure 5.6 256 bits Key, 9 chars AEAD, 12 bytes Nonce

C. Comparison between different key sizes using Nonce values as 8 bytes and 10 bytes in AES-CCM

In this section, we compare the performance by taking nonce values as 8 bytes and 10 bytes (Figures 5.7 and 5.8), respectively. The AEAD value used is constant and the key sizes were compared using a 16 KB plaintext in both cases.

We can see in the following figures that as the nonce value is increased from 8 bytes to 10 bytes, there is a vast difference in the encryption time for the key sizes 128 bits and 192 bits. The time to encrypt plaintext increases at great speed. For a key size of 256 bits, there is no substantial difference in the encryption time as compared to other key sizes. Furthermore, the time to encrypt plaintext is dependent on the nonce value; as the higher the latter becomes, the higher the former is.

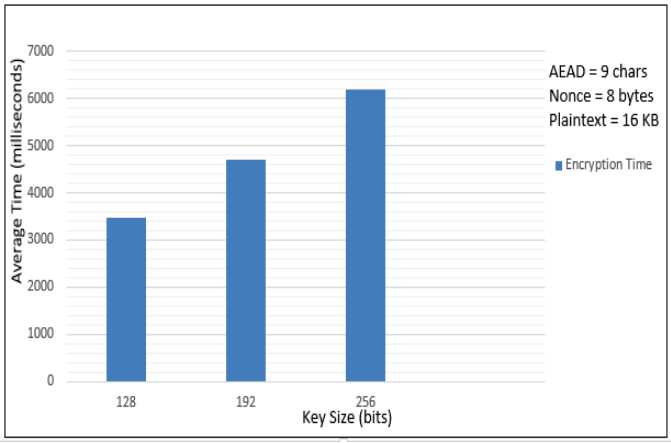


Figure 5.7 Nonce = 8 bytes, AEAD = 9 chars

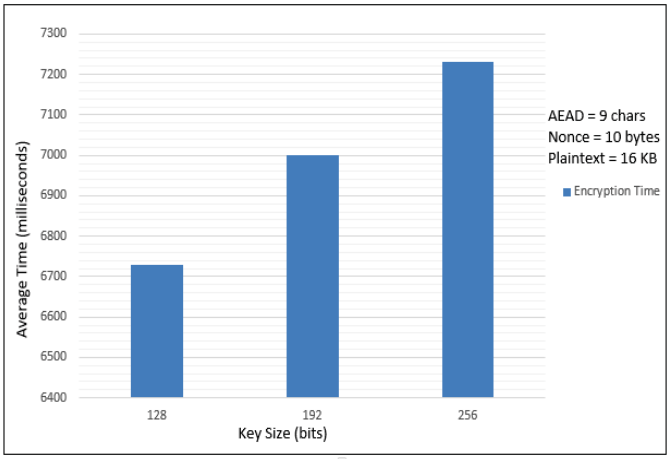


Figure 5.8 Nonce = 10 bytes, AEAD = 9 chars

D. Comparison of AES-CCM and AES-GCM

The main idea behind this paper was to compare the performance of AES-CCM and AES-GCM mechanisms in terms of time taken to encrypt the plaintext. The key size taken is 128 bits as it was kept fixed in the AES-GCM mode. As we can observe in graphs (Figures 5.7, 5.8 and 5.9), the AES-GCM has better performance than AES-CCM as it is taking less time to encrypt the plaintext.

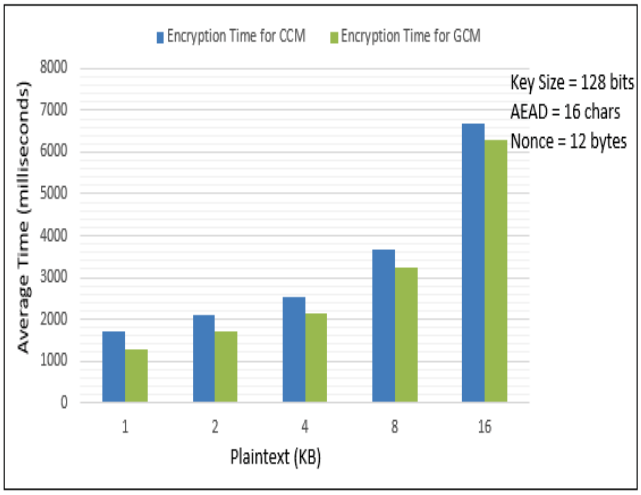


Figure 5.7 128 bits Key, 16 chars AEAD, 12 bytes Nonce

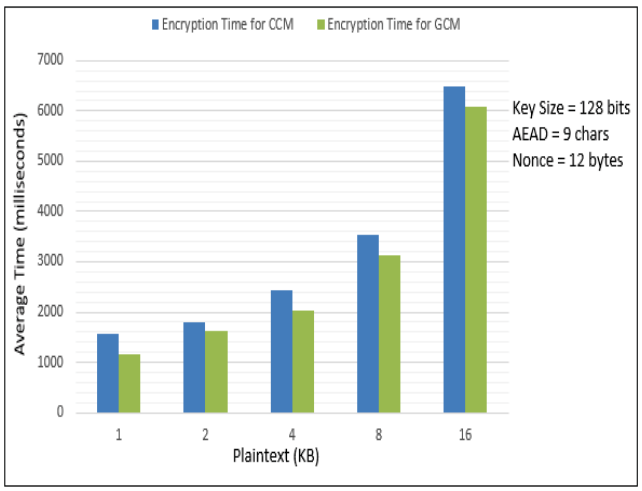


Figure 5.8 128 bits Key, 9 chars AEAD, 12 bytes Nonce

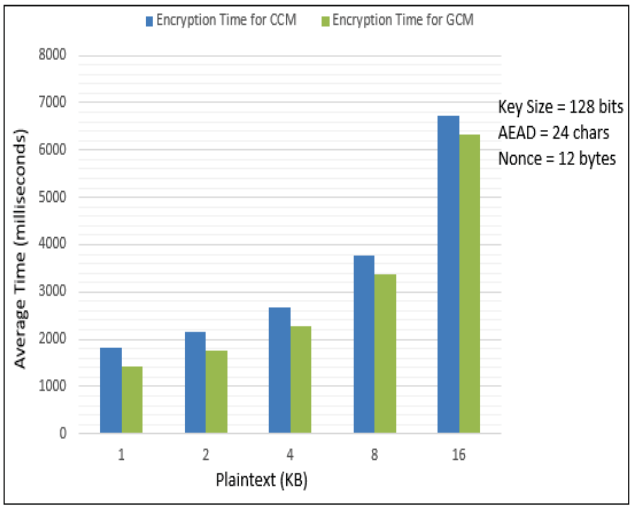


Figure 5.9 128 bits Key, 24 chars AEAD, 12 bytes Nonce

VI. Conclusion

In this paper, we have shown the implementation of an Authenticated Encryption scheme, AES-CCM, on Android application to check if this algorithm is feasible in terms of performance. We did performance analysis for AES-CCM to make comparisons by taking various parameters (key size, AEAD and nonce). These comparisons show that the performance of AES-CCM goes down when the key size, AEAD value and nonce lengths are increased. We have not noticed any major fluctuations in the encryption and decryption times of the plaintext when the key size and AEAD value were changed. However, it was noticeable that when the nonce value was increased, the encryption time rose at greater speed.

We have made another comparison between AES-CCM and AES-GCM to check which mode of operation is better performance wise. Our results show that AES-GCM is faster than AES-CCM when it comes to performance. We have come to the conclusion that the AES-GCM is more feasible to be used in applications where performance is the main concern.

VII. Acknowledgement

We respectfully acknowledge the assistance and support of Bhanuchandra Siddam and Sai Krishna Reddy Siripuram and their contribution towards the implementation and testing of the AES-GCM Authenticated Encryption mode of operation.

VIII. References

[1] M. Dworkin. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. NIST Special Publication 800-38C., 2004.

[2] Federal Information Processing Standards Publication 197. United States National Institute of Standards and Technology (NIST). November 26, 2001.

[3] National Institute of Standards and Technology Special Publication 800-38C Natl. Inst. Stand. Technol. Spec. Publ. 800-38C 25 pages (May 2004)

[4] FIPS Publication 197, Advanced Encryption Standard (AES). U.S. DoC/NIST, November 26, 2001. Available at http://csrc.nist.gov/publications/.

[5] J Daemen, V Rijmen. The design of Rijndael: AES--the advanced encryption standard. Springer Verlag, 2002.

[6] Nyberg, K., & Heys, H. (2003). Selected areas in cryptography: 9th annual international workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002: Revised papers (Vol. 2595). Berlin: Springer-Verlag.

[7] NIST Computer Security Division's (CSD) Security Technology Group (STG) (2013). "Block cipher modes". Cryptographic Toolkit. NIST.

[8] http://venturebeat.com/2013/08/01/android-reaches-massive-80-market-share-windows-phone-hits-global-high-iphone-languishes/

[9] Lemsitzer, Wolkerstorfer, Felber, Braendli, Multi-gigabit GCM-AES Architecture Optimized for FPGAs. CHES '07: Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems, 2007.

[10] "Android Studio" http://developer.android.com/sdk/index.html

[11] Genymotion – Fast and easy Android Emulation https://www.genymotion.com/

[12] Cryptography for mobile security. Mitchell, Chris 1. Security for Mobility. Ed. Chris J Mitchell. TEE Press, 2004

[13] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael" (PDF). National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013.

[14] "Information technology -- Security techniques -- Authenticated encryption". 19772:2009. ISO/IEC. Retrieved March 12, 2013.

[15] IEEE: 802.1AE-media access control (MAC) security, draft 3.5. http://www.ieee802.org/1/pages/802.1ae.html (2005)

[16] IEEE: P1619.1/d12astandard for authenticated encryption with length expansion for storage devices. http://grouper.ieee.org/groups/1619/email/bin00084.bin (2006)

[17] Viega, J., McGrew, D.: The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (EPS). http://www.faqs.org/rfcs/rfc4106.htm (2005)

[18] B. Schneider. Applied Cryptography. JohnWiley Sons, NY, 1996.

[19] RFC 4309 Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)

[20] RFC 6655 AES-CCM Cipher Suites for Transport Layer Security (TLS)

A DNA-Based Cryptographic Key Generation Algorithm

Shakir M. Hussain[1] and Hussein Al-Bahadili[1]
Department of CIS and Computer Network, University of Petra, Amman, Jordan

Abstract—*This paper presents a detail description of a new DNA-based cryptographic key generation algorithm that can be used to generate strong cryptographic key(s) for symmetric ciphering applications. The algorithm uses an initial private/secret key as an input to the Key-Based Random Permutation (KBRP) algorithm to generate a permutation of size n, which is half of the size of the required cryptographic key, and to derive four vectors of size n representing the DNA bases (A, C, G, and T) of the private key. The DNA vectors are mathematically processed using a linear formula to generate the cryptographic key. The generated bases are re-permuted using the same permutation vector and re-processed to determine new cryptographic keys, and this can be continue as much as new cryptographic keys are required. The performance of the new algorithm is evaluated in two different scenarios that demonstrate its high potential for providing high randomness cryptographic key(s). The results show that the generated cryptographic keys always have ≈0.7 entropy, and acceptable maximum and average run length for both 0's and 1's for various key-lengths and private keys.*

Keywords: DNA cryptography; DNA key generation, key generation, strong key, random permutation, KBRP.

1 Introduction

There has been a tremendous growth in the number and type of attacks that should be dealt with by data security specialists to protect sensitive valuable data, or data vulnerable to unauthorized disclosure or undetected modification, during transmission or while in storage [1]. Cryptography is a method of coding/decoding data so that it becomes unreadable or accessible by unauthorized users, which is often used to protect data during their transmission or while in storage [2]. Cryptography relies upon two main components: a cryptographic algorithm and a cryptographic key. The algorithm is a mathematical function, and the key is a parameter used by that function [3].

Cryptographic algorithms can be classified into symmetric and asymmetric algorithms. Symmetric algorithms use the same key to encrypt and decrypt data, which must be kept secret and only disclosed to authorize parties; therefore it is referred to as secret key or private key [4]. A symmetric algorithm processes data (plaintext) with the secret key to create encrypted data (ciphertext). Examples of symmetric algorithms are: DES, RC2, 3DES, AES, etc [5]. These algorithms process the secret key to generate the required cryptographic key or keys. They are extremely fast and well suited for large data encryption. However, they suffer from how to secure the secret key or how to securely exchange the secret key between different communicating parties across unsecure communication channels.

Asymmetric algorithms use two mathematically-related keys, one of these two keys is disclosed to public (hence it is referred to as public key), and the other one is kept by and only known to the user (hence it is referred to as private key) [4]. In such algorithms, data encrypted with any of these two keys can only be decrypted using the other key. Which of these keys should be used for encryption depends on the targeted security service (confidentiality or authentication). Examples of asymmetric algorithms include: Rivest-Shamir-Adleman (RSA), Diffie-Hellman (DH), Al-Gamal, etc [4, 5]. They use mathematical functions for encryption/decryption and key generation, therefore, they are relatively slow and they are mainly used for securing key-exchange over unsecure communication channels.

Symmetric algorithms can be classified into block cipher and stream cipher. A block cipher (such as DES, 3DES, AES, etc.) applies a deterministic and computable function repeatedly to encrypt a block of data at once as a group using different fixed-length cryptographic key for each cryptographic round. A stream cipher combines a plaintext stream with a cryptographic key stream in a way to produce a cipher stream, where each digit of the plaintext is encrypted one at a time with the corresponding digit of the cryptographic key stream, to give a digit of the ciphertext stream. The keys are generated using logical procedures or mathematical functions, which are normally uses some initial value or password [4, 5].

It must be well understood that lack of randomness in the logical procedures or mathematical functions of the key generators, or weak passwords, are disastrous and may lead to cryptanalytic breaks. Therefore, a number of high randomness and strong key generators have been developed [6, 7]. However, due to the exponential development in the processing power of the computing systems and the tremendous advancement of the cryptanalysis techniques, more and more powerful cryptographic and key generators are required.

Thus, in line with the growing needs for powerful cryptography, new cryptography techniques have been emerged, such as: quantum cryptography and DNA cryptography. Quantum cryptography (QC) exploits quantum mechanical properties (e.g., the counterintuitive behavior of elementary particles such as photons) to perform cryptographic tasks [8]. The best known example of this type of cryptography is quantum key distribution (QKD), which offers high-security solution to the key exchange problem rather than data encryption [9]. However, it has been discovered that QC may not be as secure as it was presumed to be, where it has been found that energy-time entanglement, which forms the basis for many systems of QC, is vulnerable to attack [10, 11].

DNA cryptography, which is working on the concept of DNA computing, is emerging as a new promising cryptographic field, where DNA is used to carry the information or to be used as an alternative data encoding approach [12]. During the last two decades, many DNA-based algorithms have been developed and used for data cryptography and cryptographic key generation [13].

In this paper, we present a detail description of a new DNA-based cryptographic key generation algorithm that can be used to generate strong cryptographic key(s) for symmetric ciphering applications. The performance of the algorithm is evaluated through two different scenarios to demonstrate its high potential for providing strong cryptographic key(s). The performance measures that are used to evaluate and compare the performance of the algorithm against other key generation algorithms include: minimum, maximum, and average run length of 0's and 1's, and entropy of key binary sequence.

This paper is divided into six sections. This section presents the main theme of this paper. The next section provides a brief background on the concept of DNA. Section 3 reviews some of the most recent and related research on DNA cryptography. The new DNA-based cryptographic key generation algorithm is given in Section 4. Section 5 presents the description of two different scenarios that are used to evaluate the randomness of generated cryptographic keys. Finally, in Section 6, conclusions are drawn and recommendations for future research are pointed-out.

2 DNA Background

Deoxyribo Nucleic Acid (DNA) is a molecule that represents the genetic material for all living organisms. It is the information carrier of all life forms, and considered as the genetic blue print of any living or existing creatures. DNA molecules consist of two long chains held together by complementary base pairs, twisted around each other to form a double-stranded helix with the bases on the inside. A DNA sequence consists of four nucleic acid bases A (adenine), C (cytosine), G (guanine), T (thymine), where A and T are complementary, and C and G are complementary [3]. The base pairing mechanism is the basis for DNA replication which is shown in Figure 1 [1].

One of the most basic attributes of the DNA strand series is that it has different orientations and each one is different from the other, e.g., TCCGAATGC is distinct from ATCGATCGC. Another basic attribute is the reverse complement, which is achieved in two stages: first is to reverse the order of the DNA strand bases, and the second is to take the complements of the reversed strands, where the complement of the base A is T and C is G and vice versa. For example, the reverse complement of AGCTAACC is GGTTAGCT [13].

The DNA sequence {A, C, G, T} is presented into binary code using a simplest coding pattern of four digits 0, 1, 2, and 3, respectively. Each digit is presented into 2-bit pattern as follows: 0 as A→00, 1 as C→01, 2 as G→10, and 3 as T→11.

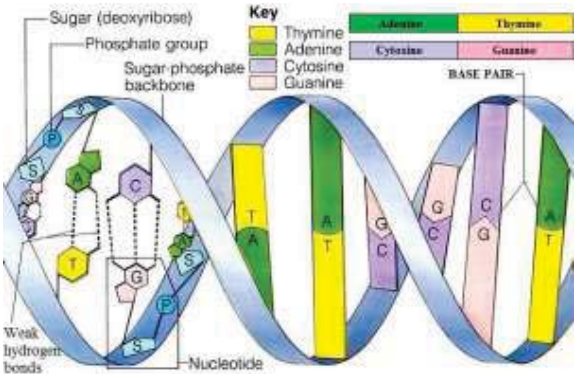


Figure 1: Structure of DNA.

The DNA sequence ACGT has 4!=24 possible pattern each of them has different numeric encoding format (e.g., 0123 for ACGT, 0132 for ACTG, 0213 for AGCT, etc.), and consequently each encoding format will have different binary representation [14].

3 Literature Review

A number of key derivation approaches have been developed throughout the years, such as: functional-based, biometric-based, voice-based, etc., a review on some of these techniques is given in [7]. However, more recently a new approach is identified, which is a DNA-based approach. DNA cryptography is a promising research approach that emerged with the evolution of DNA computing field. DNA can be used to store and transmit the information and also to perform computation. The extensive parallelism and extraordinary information density built in this molecule can be exploited for cryptographic purposes. Several DNA-based algorithms have been proposed and used in many applications, such as encryption, key generation, authentication, etc. [12]. This section briefly reviews some of the most and recent research in this area.

Ritu Gupta and Anchal Jain [15] developed a method for image encryption based on DNA computation technology. In this method, first, a secret key is generated using a DNA sequence and modular arithmetic operations. Then each pixel value of the image undergoes the encryption process using the key and DNA computation methods. The algorithm demonstrates a satisfactory computing security level in the encryption security estimating system. Zhang et al [16] proposed an image encryption algorithm based on DNA sequence addition operation. The results and security analysis show that the algorithm can demonstrate good encryption effect, and also can resist exhaustive attack, statistical attack and differential attack.

Al-Wattar et al [17] and Al-Wattar et al [18] presented alternative key-dependent DNA-based approaches for the MixColumns and ShiftRows transformations engaged in the AES algorithm, which has characteristics identical to those of the original algorithm AES besides increasing its resistance against attack. Varma and Raju [14] analyzed the different approach of DNA cryptography based on matrix manipulation and secure key generation scheme.

Liu et al [19] developed an encryption method using DNA complementary rule where piecewise linear chaotic map is used for permutation and then substitution is performed using complementary rule. An extensive review on DNA cryptography and its basic encryption techniques is presented in [12, 20].

4 The Proposed Algorithm

A private key may be considered as a living creature with a genetic blueprint (i.e., DNA) that can be derived and used as a cryptographic key in single cryptographic key symmetric algorithms. The DNA can be used to derive further cryptographic keys for multi cryptographic key symmetric algorithms. For examples, DES requires sixteen 48-bit keys and AES requires ten 128-bit keys) [4, 5].

The proposed DNA-based cryptographic key generation algorithm can be summarized as follows:

1. A private key is used to generate a permutation P of size n, where n is half of the size of the required cryptographic key (k) using any permutation generation algorithm. In this work we use the KBRP algorithm [21]. The KBRP method derives one permutation of size n out of $n!$ possible permutations for any given private key or password. For k-bit key, n=k/2 (e.g., for the DES, since k=56, then n=28).
2. The permutation P is used to generate the DNA-based cryptographic key as follows:
   a. Convert the n different values of the permutation P to their equivalent binary value (one byte each).
   b. Convert each two consecutives bits to an integer value between 0 to 3.
   c. Store these integer values in a vector V of size $4n$.
   d. Split the vector V into four vectors (V_1, V_2, V_3, and V_4) each of size n.
   e. Permute the vectors (V_1, V_2, V_3, V_4) using the permutation P to produce permuted vectors (PV_1, PV_2, PV_3, PV_4.
   f. For a single cryptographic key application, the n elements of the DNA can be calculated as:

For u = 1 To n

$$DNA(u)=(PV_1(u)+PV_2(n-u+1)+PV_3(u)+PV_4(n-u+1)) \% 4 \quad (1)$$

Next u

For a multi cryptographic key application, the DNA bases can be calculated as:

For v = 1 to m

For u = 1 To n

$$DNA(u,v)=(PV_1(u)+PV_2(n-u+1)+PV_3(u)+PV_4(n-u+1)) \% 4 \quad (2)$$

Next u

Permute PV_1, PV_2, PV_3, and PV_4 using the permutation P

Next v

Where m represents the number of required cryptographic keys, for example 16 cryptographic round keys, one for each round of the DES algorithm, or 10 cryptographic round keys for the AES algorithm. Each element of the DNA vector will have n values, each value lies between 0 to 3, which can be converted to DNA bases.

   g. Convert each DNA base to its 2-bit equivalent value (A as 0→00, C as 1→01, G as 2→10, and T as 3→11). This will yield the k-bit cryptographic key(s).

In this method the DNA components are randomly distributed over the DNA-based generated key without any previous knowledge about the occurrence of each DNA component.

5 Performance Evaluation

In this paper, in order to demonstrate the tremendous potential and evaluate the statistical performance of the new DNA-based cryptographic key generation algorithm, we develop two scenarios. In the first scenario (Scenario #1), we determine the statistical parameters (e.g., minimum, maximum, and average run-length of 0's and 1's, and entropy) for a number of cryptographic keys generated by the new DNA-based algorithm using different private keys; namely, "Computer", "Ad-Hoc", and "CDMA&2000". Different cryptographic key sizes are generated using the same private set of keys (e.g., 64, 126, 256, 512, and 1024 bits). The generated cryptographic keys demonstrate excellent statistical features as shown in Table (1).

In particular, the results show that the generated cryptographic keys always have the maximum acceptable entropy, a controlled run length for both 0's and 1's for all key lengths, and an acceptable average run length. For example, for the three different private keys, the maximum run length for 1's in a key of 1024-bit is 14, which is equivalent to 1.4% of the total key length.

In the second scenario (Scenario #2), we use the new algorithm to generate the cryptographic (round) keys for the DES algorithm (16 rounds), and compare the statistical parameters of the generated keys against those generated using the standard DES key generator [4]. The results are presented in Table (2) for the new algorithm and in Table (3) for the DES key generator. The private key using in this scenario is "Computer" to generate 16 48-bit round keys.

It can be clearly seen from Tables (2) and (3) that the new algorithm provides promising statistical result on the key in terms of entropy, minimum, maximum, and average run-length for both 0's and 1's. The features are very competitive with the standard DES key generator.

Table (1) – Scenario #1.								
Private-Key	Cryptographic Key Length	Run-Length for 0			Run Length for 1			Entropy
		Min.	Max.	Avg.	Min.	Max.	Avg.	
Computer	64-bit	1	8	2.375	1	3	1.733	0.675
	128-bit	1	9	2.567	1	5	1.645	0.672
	256-bit	1	7	2.200	1	6	2.033	0.693
	512-bit	1	11	2.205	1	7	2.008	0.692
	1024-bit	1	8	1.909	1	11	2.122	0.692
Ad-Hoc	64-bit	1	4	2.063	1	5	2.067	0.693
	128-bit	1	6	1.968	1	9	2.094	0.692
	256-bit	1	5	1.955	1	6	1.838	0.693
	512-bit	1	10	2.235	1	9	2.067	0.692
	1024-bit	1	9	1.962	1	8	1.924	0.693
CDMA&2000	64-bit	1	5	2.467	1	5	1.800	0.681
	128-bit	1	6	1.935	1	6	2.194	0.691
	256-bit	1	5	1.625	1	6	1.931	0.689
	512-bit	1	9	1.963	1	6	1.830	0.693
	1024-bit	1	7	2.107	1	14	2.098	0.693

Table (2) – Scenario #2 -Statistical parameters using the DNA-base key generator (Private key is Computer)							
Round	Run Length for 0			Run Length for 1			Entropy
	Min.	Max.	Avg.	Min.	Max.	Avg.	
1	1	6	2.063	1	3	1.438	0.677
2	1	6	2.071	1	6	1.800	0.693
3	1	7	2.615	1	4	1.692	0.670
4	1	5	2.267	1	4	1.571	0.670
5	1	4	1.625	1	4	2.000	0.691
6	1	6	2.200	1	5	1.438	0.677
7	1	5	2.133	1	5	1.600	0.683
8	1	7	2.357	1	4	1.533	0.677
9	1	6	2.143	1	7	1.857	0.691
10	1	6	2.385	1	4	1.786	0.687
11	1	5	1.929	1	5	2.071	0.693
12	1	6	2.308	1	3	2.000	0.691
13	1	5	2.429	1	6	1.692	0.670
14	1	4	1.944	1	3	1.235	0.662
15	1	4	2.231	1	8	2.250	0.693
16	1	4	1.722	1	3	1.389	0.687

Table (3) – Scenario #2 - Statistical parameters using the DES key generator (Private/secret key is Computer)							
Round	Run Length for 0			Run Length for 1			Entropy
	Min.	Max.	Avg.	Min.	Max.	Avg.	
1	1	9	2.400	1	4	2.400	0.683
2	1	5	1.769	1	5	1.923	0.677
3	1	8	2.300	1	7	2.273	0.677
4	1	4	1.909	1	4	2.250	0.662
5	1	3	1.571	1	3	1.733	0.670
6	1	4	1.769	1	6	1.923	0.677
7	1	8	2.556	1	6	3.125	0.677
8	1	9	1.917	1	6	2.083	0.677
9	1	3	1.438	1	5	1.667	0.677
10	1	4	1.833	1	6	2.364	0.670
11	1	6	1.846	1	3	1.846	0.683
12	1	4	1.769	1	6	1.923	0.677
13	1	7	1.643	1	4	1.786	0.677
14	1	8	2.273	1	4	1.917	0.687
15	1	6	1.917	1	5	2.083	0.677
16	1	8	2.889	1	5	2.444	0.691

5 Conclusions

This paper presents a detail description of a new DNA-based cryptographic key generation algorithm that can be used to generate strong cryptographic key(s) for symmetric ciphering applications. The algorithm is used in two different scenarios to demonstrate its high potential for providing strong cryptographic key(s). The two scenarios show that the generated cryptographic keys always have an 0.7 entropy, an optimum run length for both 0's and 1's for all key lengths, and an acceptable average run length. For 48-bit cryptographic key, it presents 14% maximum run-length for 0's and 9% for 1's, and average run-length of 4% for both 0's and 1's. These parameters decrease with increasing key length.

This algorithm is at its early stage of development and it is open up an area of interesting research. For example: (1) Develop and perform more evaluation procedures and techniques, and (2) use the algorithm as a cryptographic key generator for the standard symmetric encryption algorithms (e.g., DES, 3DES, AES, IDEA, etc.) and compare the statistical randomness test of the produced ciphertext against using the standard key generator of each of these algorithms.

REFERENCES

[1] M. Zhang, M. X. Cheng, and T. J. Tarn. A mathematical formulation of DNA computation. IEEE Transactions on NanoBioscience, Vol. 5, No. 1, pp. 32-40, 2006.

[2] P. Saxena, A. Singh, and S. Lalwani. Use of DNA for computation, storage and cryptography of information. International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 3, Issue 2, pp. 2278-3075, 2013.

[3] Bibhash Roy, Gautam Rakshit, Ritwik Chakraborty. Enhanced key generation scheme based cryptography with DNA logic. International Journal of Information and Communication Technology Research, Volume 1, No. 8, December 2011.

[4] B. A. Forouzan. Introduction to Cryptography and Network Security. McGraw-Hill (International Ed.), 2008.

[5] W. Stallings. Cryptography and Network Security: Principles and Practices. Prentice Hall (6th Ed.), 2014.

[6] E. Barker and A. Roginsky. Recommendation for Cryptographic Key Generation. NIST Special Publication 800-133, 2012.

[7] S. M. Hussain and H. Al-Bahadili. A password-based key derivation algorithm using the KBRP method. American Journal of Applied Sciences, Vol. 5, No. 7, pp. 777-782, 2008.

[8] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. Report on post-quantum cryptography. National Institute of Standards and Technology Internal Report, NISTIR 8105, February 2016.

[9] A. Mink, S. Franke, and R. Perlner. Quantum Key Distribution (QKD) and Commodity Security Protocols: Introduction and Integration. International Journal of Network Security & Its Applications (IJNSA), Vol. 1, No. 2, pp. 101-112, July 2009.

[10] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. Nature Photonics, Vol. 4, pp. 686–689, 2010.

[11] J. Jogenfors, A. M. Elhassan, J. Ahrens, M. Bourennane, and J. A. Larsson. Hacking the bell test using classical light in energy-time entanglement–based quantum key distribution. Science Advances, Vol. 1, No. 11, 2015.

[12] T. Mandge and V. Choudhary. A review on emerging cryptography technique: DNA cryptography. International Journal of Computer Applications (IJCA), Vol. 13, pp. 9-13, February 2013.

[13] B. B. Raj and V. Panchami. DNA-based cryptography using permutation and random key generation method. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 5, pp. 263-267, July 2014.

[14] P. S. Varma, K. G. Raju. Cryptography based on DNA using random key generation scheme. International Journal of Science Engineering and Advance Technology (IJSEAT), Vol. 2, Issue 7, pp. 168-175, July, 2014.

[15] Ritu Gupta and Anchal Jain. A new image encryption algorithm based on DNA approach. International Journal of Computer Applications, Vol. 85, No. 18, pp. 27-31, January 2014.

[16] Q. Zhang, L. Guo, X. Xue, and X. Wei. An image encryption algorithm based on DNA sequence addition operation. Proceedings of the 4th International conference on Bio-Inspired Computing (BIC-TA '09), pp. 1-5, Beijing, China, 16-19 October 2009.

[17] A. H. Al-Wattar, R. Mahmod, Z. A. Zukarnain, and N. Udzir. A new DNA based approach of generating key dependent MixColumns transformation. International Journal of Computer Networks & Communications (IJCNC), Vol. 7, No. 2, pp. 93-102, March 2015.

[18] A. Al-Wattar, R. Mahmod, Z. Zukarnain, and N. Udzir, "A new DNA based approach of generating key-dependent ShiftRows transformation. International Journal of Network Security and Its Applications (IJNSA), Vol.7, No.1, January 2015.

[19] H. Liu, X. Wang, and A. Kadir. Image encryption using DNA complementary rule and chaotic maps. Applied Soft Computing, Vol. 12, pp. 1457–1466, 2012.

[20] Pierluigi Paganini. The future of data security: DNA cryptography and cryptosystems. Retrieved from http://securityaffairs.co/wordpress/33879/security/dna-cryptography.html on 20th February 2015.

[21] S. M. Hussain and N. M. Ajlouni. Key-based random permutation (KBRP). Journal of Computer Science, Vol. 2, No. 5, pp. 419-421, 2006.

SESSION

POSTER PAPERS

Chair(s)

TBA

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

s-Tor-y on User Acceptance of PETs.

Marta Piekarska[1], **Miguel Rios Quintero**[2]
[1]Security in Telecommunications, Technische Universität Berlin, Berlin, Germany
[2]IP-Based Assesment, Technische Universität Berlin, Berlin, Germany

Abstract— *Today every user has a plethora of devices to choose from, depending on the task they want to perform. However every move they make leaves a trace on the internet. Global surveillance is so common we learned to accept it and not even try to fight against it. There are existing solutions that help protect users, however little research has been done on the usability and transparency of these tools. We try to convince people that it is better to sacrifice some of the convenience for better privacy, instead of evaluating how to change existing mechanisms to fit their needs better. In our research we would like to gain insights on how users across countries differ in their perception of privacy and security and what is their willingness to accept Privacy Enhancing Technologies on the example of Tor running on a mobile device. To this end we plan to conduct a two phase study consisting of an online background survey and a multi group lab test.*

Keywords: Mobile Privacy, Mobile Security, Privacy Enhancing Technologies, User-Centric Research, Usability of Privacy, Quality of Privacy Experience

1. Introduction

Today global surveillance is a fact. Almost 6 billion of us have a device in pocket that traces what we do, where we do it and how we do it. Maintaining personal privacy is hardly possible. Some researchers and Research and Development Centers in the Industry are trying to come up with Privacy Enhancing Technologies (PETs). Tools like PGP email encryption, OTR-supported chats, encrypted VoIP applications are created to secure data and communication. While focusing on airtight security, however, we all tend to forget about the basic problem - users will not change their technology habits to accommodate the PETs. They will not sacrifice the usability of their fancy smartphones for a promise of a more private life. PETs do not offer users more features or a better experience, so the best way to drive adoption is by making PETs both transparent and automatic.

In this research we want to focus on one of the best anonymizing tools - Tor. We would like to understand the big picture first: what is the cultural difference in perception of security and privacy between Germany and the United States. Next we move to the acceptability and usability of Tor on mobile devices. We want to study both groups of people unaware of what service is running in the background and ones that do know. Finally we will discuss methods to improve Tor integration into the mobile world and give an outlook to the future work on development of more transparent PETs on mobile platforms.

2. Background

Privacy enhancing technologies (PET) is a general term that is used to name any application or mechanism that helps allows to protect user privacy and not reveal their Personally Identifiable Information online. Examples of existing PETs include communication anonymizing solutions, shared bogus online accounts, enhanced privacy ID and services that protect access to personal data. Tor is an example of a free software and an open network that belongs to the first group. It helps to defend against traffic analysis by implementing the onion protocol that distributes the online transactions over several places on the Internet, so that there is no single link to the origin of the request. Instead of taking a direct route from source to destination, data packets on the Tor network take a random pathway through several relays that cover your tracks so no observer at any single point can tell where the data came from or where it's going [1].

3. Hypothesis and Research Questions

Users will only adapt transparent Privacy Enhancing Technologies that are fully invisible, they do not want to know there is something running on their device. With the notion of Security and Privacy tools comes a certain attitude of perceiving them as ?scary? and ?highly technical?. One of the best examples of that is Tor. Tool that preservers anonymity, is also considered as having extreme impact on usability - it is said to be slow, complicated and connected to hackers and dark matters. With this study we want to verify this approach - that Tor on a mobile phone can be used as part of the Privacy by Default approach, as long as we account for the drawbacks that Tor introduces in the design of the system.

We would like to look at how does running privacy preserving tools influence the annoyance with the network performance when doing everyday tasks. What follows is if

people change their attitude and perception of usability if they know that there is a privacy preserving tool running on their device, moreover if there is any difference in just any tool and one they have heard or read about. Effectively, if in this case knowledge and education becomes a curse. Finally, we would like to understand the threshold - what is the minimum network quality for people to be satisfied with the performance of their devices when Tor is running in the background.

In addition we would like to understand what are the differences in the knowledge and approach to security and privacy between Germany and the United States. We choose these two countries as they are known to be on the two sides of the spectrum, where the people in the former are obsessed with their right to privacy, while in the latter - need to overshare. We are curious to see what and if there is a difference in the attitude and willingness of users to adapt PETs in the two countries.

4. Planned Study

Our study consists of two phases. In fist phase we plan to do a parallel survey online in Germany and in the United States that will verify the following aspects of users' perception of security and privacy:

- definition of security and privacy
- perception of importance of Security and Privacy protection on mobile devices
- attitude towards security and privacy
- knowledge of technical aspects - how can they protect themselves, what tools are available today.

Especially understanding the differences in the cultural influence on the definition of privacy and security and their importance will be very interesting to us. We see value in analyzing how to build tools that are fitting the needs of the target groups and not the opposite - trying to square the users into our believes of what are the best solutions or them. After the first phase of the study we will move to the Laboratory evaluation. This will consist of the measurement of the three aspects, as described further. We discuss the methodology in the next section.

4.1 Network Quality

In this setting we would like to analyze how does the quality of the network influence the perception of the usability of the device [2]. We will be manipulating the quality of the connection speed recreating an environment of a typical 2G, 3G and High Speed Internet Connectivity. We will then ask the users to fill out a questioner to express their opinion on the quality of experience and annoyance [3]. This part of the research will contribute to the academia by showing what are the minimum requirements for network to consider running Tor or similar services. By understanding the impact of network quality on user perception of Tor on mobile devices we will be able to improve Tor mobile application and set guidelines for when to enable it.

4.2 Knowledge

This aspect will be devoted into looking at how knowledge about a Privacy Enhancing technology, and possibly is type, influences user expectation, and perceived quality of experience. Some subjects will be asked to perform tasks with no information of Tor being enabled on their devices. Another group will be running Tor and will be informed about it. Finally a control group will not be running Tor at all. We are also considering verifying the Placebo effect - how does perception changes if we tell the participants they are using Tor while it is not enabled? We believe that this part of the study will contribute by showing how much of trouble with adapting PETs has to do with communication and the fact that technology is often considered "scary" or "unapproachable". We have not yet seen studies that combine the fields of Computer Science and Communication and Psychology, yet it it important to have a good understanding of users' mindsets in order to build good tools.

4.3 Activity

The final aspect that the Lab study will consist of is performing tasks that users normally would on their mobile phones. These include:

1) Checking their email,
2) having a short Whatsapp or iMessage conversation,
3) Watching a you tube video
4) Interacting with Facebook both online and through the Facebook app that has Tor integrated,
5) Doing banking/shopping/news reading - whatever they see as most natural.

The reason why we ask them to perform these tasks is to recreate circumstances as close to natural environment as possible. This way we can get their assessment more objectively. By doing what they would normally do on their devices they will be able to compare if and how their devices change their behavior and if it has significant impact on usability or quality of experience.

5. Measurements

The online survey will be done with the use of the in-house tools build by our research group, Crowdee [4]. For the lab study we will be recruiting participants in Germany through various media and tools available, aiming for a well distributed group of about 100 participants. They will be given Android or iOS phones, depending on what system they are primarily using. The mobile version of Tor that will be running on the devices is Orbot [1] with modifications that will allow us to monitor the following aspects of the performance: Packet loss, bandwidth, round trip time, time

[1] https://guardianproject.info/apps/orbot/

it takes to establish initial circuit, usage of the network resources to establish the connection, usage of network resources to download the available circuits, usage of the network resources when idle. In the lab we will have a controlled virtual network where we will be adjusting the speed of the connection according to the required level.

For the user-study we plan to use open-ended questioners to measure the satisfaction level and the acceptability of the performance drop. We will also be using survey techniques based on [5] to verify the knowledge and perception of security and privacy during phase one of the study. The author of the paper is presenting a pilot study that verifies a model he proposes that includes users' general privacy concern, the environmental influences on these worries and , and organization- and technology-specific attributes of it. We think that this is a very interesting starting point for creating our own model of studying user perspective on Privacy Enhancing Technologies. We will divid the lab group into four groups of awareness, as described in Section 4.2. All of them will be asked to perform tasks described in Sectionactivity in three different settings of network performance. The test subjects will be given a financial reward for participating in the test.

6. Conclusions

Looking at how users adapt Privacy Enhancing Technologies is an important task for researchers. It is crucial to understand the mental model of people in order to create tools that are useful and easy to adapt for them. Otherwise we will continue to develop solutions that are great from technological standpoint of view but not used in reality. It is also valuble to understand the influence of communication about already developed PETs. Moving forward this will help researches shape the introduction of their ideas in a better way. We observe a similar problem with Tor as with Bitcoin, where more negative communication gets to the media changing how people perceive tools that can be used for very good reasons.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251375.1251396

[2] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying skype user satisfaction," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '06. New York, NY, USA: ACM, 2006, pp. 399–410.

[3] F. Kuipers, R. Kooij, D. De Vleeschauwer, and K. Brunnström, "Techniques for measuring quality of experience," in *Proceedings of the 8th International Conference on Wired/Wireless Internet Communications*, ser. WWIC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 216–227.

[4] B. Naderi, I. Wechsung, T. Polzehl, and S. Möller, "Development and validation of extrinsic motivation scale for crowdsourcing micro-task platforms," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, ser. CrowdMM '14. New York, NY, USA: ACM, 2014, pp. 31–36. [Online]. Available: http://doi.acm.org/10.1145/2660114.2660122

[5] A. Morton, "Measuring inherent privacy concern and desire for privacy - a pilot survey study of an instrument to measure dispositional privacy concern," in *Social Computing (SocialCom), 2013 International Conference on*, Sept 2013, pp. 468–477.

Submitted as Poster

A Vision for Intrusion Analysis and Digital Forensics in Cloud

Dr. Kazi Zunnurhain
Assistant Professor
Northern Kentucky University
Highland Heights, KY 41099
+1 859 572 4753
zunnurhaik1@nku.edu

ABSTRACT
Cloud computing is gaining in popularity and complexity. As more companies move their platforms, services and infrastructure to the cloud, security becomes an increasingly important and daunting task. This paper is aimed at exploring improvements to secure authentication and encryption as well as focusing on the changes to intrusion analysis and digital forensics when moving to the cloud, with a future intention to propose a consolidated platform to support rigid authentication and encryption in cloud with support of digital forensics technologies.

Keywords
Cloud computing, digital forensics, authentication, encryption.

1. INTRODUCTION
Cloud computing has become a very widespread and popular technology. Forbes reported that cloud computing was a $56.6 billion dollar industry in 2014 and predicts continuous growth in the future [1]. Cloud computing has been defined in several ways and one of the most common definitions is from the National Institute of Standards and Technology (NIST): "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*."

Authentication, encryption, and intrusion analysis are important aspects for the future of cloud security. The 2015 data breach investigations report published by Verizon Enterprises noted that credentials were used to gain access to over half of the systems breached in the year 2014 [7]. As threats to computer systems and the cloud are evolving, so should our methods of securing them. In Section 2, alternate forms of authentication are presented as a method for securing the cloud. In Section 3 the need for cloud security for forensic intrusion analysis is introduced. Section 4 discusses the future directions and concludes the paper with our future intension.

2. AUTHENTICATION
Authentication is one of the most prominent feature of security triads (among Confidentiality, Integrity and Authentication). With authenticity the privilege of a user is determined in a system. In this section we will analyze different approaches of authentication mechanism, such as: improving passwords, alternatives of authentication.

2.1 Improving Passwords
Passwords are hard for humans to remember and the typical text based passwords are too easy for computers and our adversaries to guess. There are measures that system administrators can consider to increase the complexity, such as how often the user must change the password or limiting the reuse of previous passwords. One study showed that over 30% of participants wrote the password down on a piece of paper or stored it electronically in clear text [8]. Passwords are also a single point of failure in most systems; if hackers learn a user's password they can access everything the user is authorized to access on that system.

2.2 Authentication Alternatives
There are multiple authentication methods aimed at replacing or augmenting password usage to increase security. These methods individually have not been proven as feasible replacement for passwords but if combined with passwords or other authentication methods, can provide a much higher level of security than passwords alone. There are several categories of authentication methods that exist today.

2.2.1 Single Sign On
Federated single sign on is an authentication method that allows users to create a single identity through an identity management system and the other service providers trust the identity provider (IDP) to authenticate the user and validate credentials. An example of this is Google OAuth 2.0 or Facebook Connect where users login with their Google or Facebook credentials and connect to other websites or applications with similar identity [9].

2.2.2 QR code
QR code authentication uses either a smartphone or a piece of paper and a webcam, the user proves identity through possession of the token on screen or on paper. On a smart phone a user identifies himself using an app that scans a QR code to validate the user for login based on an identity setup on first use of the phone [9].

2.2.3 Biometrics
Biometric authentication uses a human related characteristic to identify and authenticate users. Common forms are

fingerprints, facial recognition, voice recognition, and retinal scanners. Unique characteristics of each user are used to initially identify the user and allow them to use that characteristic to authenticate them in the future.

3. FORENSIC INCIDENT ANALYSIS

In conjunction with authentication and encryption, intrusion analysis is an important topic of cloud computing security. The NIST definition of cloud computing forensic is "*the application of scientific principles, technological practices, and derived and proven methods to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation, and reporting of digital evidence*" [2]

The 2015 data breach investigations report published by Verizon Enterprises noted 79,790 reported security incidents in the year 2014 [7]. Cloud computing forensics is in its infancy, and different environments introduced by the cloud created issues for the traditional forensic process. The issues facing cloud forensics are highly documented in many studies, but currently there are only theoretical architectural framework and auditing solutions to address the issues of cloud computing forensics [2], [3].

In an effort to create standard for forensic analysis NIST researched, analyzed, and summarized seventy studies on cloud computing forensics, some of those categories will be discussed in the following sections.

3.1 Cloud Architecture

One of the biggest challenges to cloud forensics is the cloud architecture itself. Each environment is completely different from the other. The physical amount of data that could reside in the cloud can pose a problem for an investigator because if collected data needs to be stored. The data itself can be virtual, volatile, or stored on a hard drive, hence the removal of data will become an issue. If the space is reused then the questions of access to the data, imputed evidences, and data modification are raised.

Security needs to be defined with boundaries in multi-tenant environments. This security would help an investigator from breaching the confidentiality of the other users residing in the same environment. The cloud also needs access control security to assist the investigator in documenting anyone who had access to the data as well as the possibility of contaminated data.

3.2 Collection of Data

Data collection in the cloud environment is also affected by the architecture. Existing forensic techniques would seize storage media or computers and peripherals, but in a cloud environment that is not possible. Locating and isolating all of the data and its copies is a monumental task within itself because of the global aspect of the cloud and since only a real time collection can be performed. Real time data collection and recovery of deleted data has issues due to the volatility of the data in the cloud environment, and the use of live forensic tools cause environmental changes in the system [5].

3.3 Laws and Jurisdiction

The global nature of a cloud environment creates a challenge on how to bring all of the contributors involved in the cloud environment to work together in the collection and validation of digital evidence. This collaboration can cross global and legal jurisdictional boundaries and lead not only to technical challenges, but also to legal and organizational challenges [2].

Security in the cloud would have to include the creation of international laws that will work equally across the global infrastructure of the cloud. It would also have to encompass a form of timely international communication and cooperation so that forensic investigations and requests for data are consistent.

4. FUTURE DIRECTIONS AND CONCLUSION

Research is continuing in the areas of operating system including encryption and memory encryption to prevent security breaches such as the cold boot attacks [15]. One of the technologies called PRIME is being developed to support multiple RSA keys. At the software level there is a plan to redesign system programming languages to allow users more control over the register allocation process [12]. Solutions like PRIME, which uses an infrastructure that protects RSA private keys, and software solutions work to prevent stacks such as the cold boot attack [14]. These approaches will not solve all potential security issues, but can be used in conjunction with other types of security to improve the security.

Currently the development of standards and guidelines for cloud computing forensics is being led by NIST at the request of the Federal Chief Information Officer. NIST is working with the private sector to create the standards so they are cost-effective while meeting forensic requirements. This work has to be thorough, timely, and adaptable so it does not become obsolete as technology advances like the current set of forensic standards did [2]. The standards also have to be applicable to a global environment while not putting a financial or performance burden on the suppliers of cloud resources. Finally, standards also have to ensure personally identifiable information and proprietary information.

In future we would like to propose a concrete model to support authentication with the replacement of passwords for clients in cloud system with a promise to sustain integrity. Most of the IDS and Password authentication protocol do not consider the laws and jurisdictions based on diverse geographic location. Hence in our extended model we will support the SLA to be consolidated based on local laws and restrictions to avoid unexpected breaches or disputes. Future of cloud should be globally supported throughout any application with efficient security and authentication.

5. REFERENCES

[1] Quick, D. and Choo, K.K.R., 2013. Forensic collection of cloud storage data: Does the act of collection result in changes to the data or its metadata? *Digital Investigation*, *10*(3), pp.266-277.

[2] Dykstra, J. and Sherman, A.T., 2011, January. Understanding issues in cloud forensics: two hypothetical case studies. In *Proceedings of the Conference on Digital Forensics, Security and Law* (p. 45). Association of Digital Forensics, Security and Law.

[3] Almulla, S.A., Iraqi, Y. and Jones, A., 2014. A state-of-the-art review of cloud forensics. *Journal of Digital Forensics, Security and Law*, *9*(4), pp.7-28.

[4] Lallie, H.S. and Pimlott, L., 2012. Applying the ACPO Principles in Public Cloud Forensic Investigations. *The*

Journal of Digital Forensics, Security and Law: JDFSL, *7*(1), p.71.

[5] Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F. and Egelman, S., 2011, May. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2595-2604). ACM.

[6] Bonneau, J., Herley, C., Van Oorschot, P.C. and Stajano, F., 2012, May. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on* (pp. 553-567). IEEE.

[7] Blass, E.O. and Robertson, W., 2012, December. TRESOR-HUNT: attacking CPU-bound encryption. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 71-78). ACM.

[8] Garmany, B. and Müller, T., 2013, December. PRIME: private RSA infrastructure for memory-less encryption. In *Proceedings of the 29th Annual Computer Security Applications Conference* (pp. 149-158). ACM.

[9] Simmons, P., 2011, December. Security through amnesia: a software-based solution to the cold boot attack on disk encryption. In *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 73-82). ACM.

[10] Kaushik, A. and Naithani, S., 2014. Software Solution: Against Cold Boot Attack.

[11] Henson, M. and Taylor, S., 2014. Memory encryption: a survey of existing techniques. *ACM Computing Surveys (CSUR)*, *46*(4), p.53.

[12] Götzfried, J. and Müller, T., 2014. Mutual authentication and trust bootstrapping towards secure disk encryption. *ACM Transactions on Information and System Security (TISSEC)*, *17*(2), p.6.

[13] Bangor, A., Kortum, P. and Miller, J., 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, *4*(3), pp.114-123.

[14] Ruoti, S., Kim, N., Burgon, B., Van Der Horst, T. and Seamons, K., 2013, July. Confused Johnny: when automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security* (p. 5). ACM.

[15] Baldini, G., Botterman, M., Neisse, R. and Tallacchini, M., 2016. Ethical Design in the Internet of Things. *Science and Engineering Ethics*, pp.1-21.

A Denied-Events based Detection Method against SSH Brute-force Attack in Supercomputing Service Environment

Jae-Kook Lee[1], **Sung-Jun Kim**[1], **and Taeyoung Hong**[1]
[1]Department of Supercomputing Infrastructure, KISTI, Daejeon, KOREA

Abstract— *The brute-force attack is one of general cyber security threats in supercomputing service environment using a secure shell (SSH) protocol. First we analyzed SSH brute-force attacks had been detected through the log file parsing method of servers in the KISTI. We found that SSH brute-force attacks are classified '1:1', '1:N' or 'N:1' types of attack between source and destination IP address. And the duration of attacks that is generally the time it takes to execute attacks keeps enough long time. In this paper, we propose a SSH brute-force attack detection method using denied-events of firewalls and evaluate the effectiveness of the method. The analysis results show that our method filter beforehand by 46% on average that the attack traffic flow to active servers.*

Keywords: Brute-force attack; Supercomputer; Security; SSH, Attack Detection

1. Introduction

The brute-force attacks is the major security threat against remote services such as SSH [1]. The SSH brute-force attack attempts to gain abnormal access by guessing a user account and password pair. The SANS Institute called brute-force attacks against SSH to "the most common form of attack to compromise servers facing the internet [2]." And on May 2015, MaAfee Labs show top network attacks in Q1 [3]. A brute-force attack has been ranked in top 2. Thus it is one of classical security threats but still persists for several decades.

In this paper, we analyze SSH brute-force attacks are detected in the KISTI supercomputing service environment using the log file parsing method of servers [4]. In [4], parsed log files are clustered together in the same Src. IP or user account. If the number of failed logs in a group by Src. IP or user account exceeds the threshold, then it is detected a SSH brute-force attack. Analysis result shows that SSH brute-force attacks are classified '1:1', '1:N' or 'N:1' types of attack between Src. and Dst. IP. And the duration of attacks that is generally the time it takes to execute attacks keeps enough long time.

In order to reduce the load of active servers from long time attacks, we propose a SSH brute-force attack detection method using denied-events of network firewalls. In order to detect a SSH brute-force attack, this approach classifies by Src. IP or Dst. IP and detects attacks by checking whether dropped events above a pre-defined threshold. If there are attacks, the Src. IP is inserted to the new firewall rule automatically. We evaluate the effectiveness of the method in our environment. The result show that the attack traffic directed to active servers are reduced.

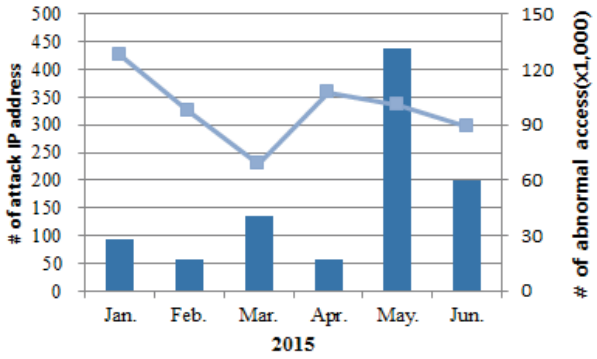


Fig. 1: Detected SSH brute-force attacks and abnormal accesses by [4] (monthly).

The rest of this paper is organized as follows: Section 2 analysis SSH brute-force attacks that are detected using the log file parsing method of servers in the KISTI. And we provides detail of SSH brute-force attack detection methods using denied-events of a firewall and evaluate the effectiveness of the method in Section 3. Finally, we conclude in Section 4.

2. SSH brute-force attacks analysis

We could detect to 983 SSH brute-force attacks using the fail-log based detection method was proposed in [4] for 6 months. If a number of accesses have same SSH process ID then we count it as one access log. Fig. 1 shows monthly SSH brute-force attack IP and distribution of abnormal access attempts detected by [4]. In Jan., Feb. and Apr., there are less SSH brute-force attack IP than other month. But there are too many abnormal accesses against SSH remote service. It is classified '1:N' type attacks such as DoS (Denial of Service). On the contrary, in May, there are many attack IP but on the contrary there are less access attempts with SSH brute-force attack than others. It is classified 'N:1' type attack such as distributed DoS attacks.

Attack duration is the time difference from the point of first traffic occurrence and the point of final traffic

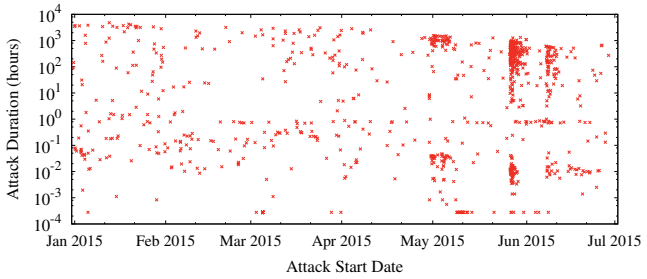


Fig. 2: Attack duration of SSH brute-force attacks.

Table 1: Events of the KISTI supercomputer firewalls

	Data
Date	2015.1.1 - 2015.6.30
Total log file size	231.96 Gbytes
Total firewall events	1,274,163,692

occurrence within the relevant period. Fig. 2 shows the distribution of SSH brute-force attacks. Fig. 2 is a log-linear plot where the horizontal axis represents the first access time and the vertical axis represents the attack duration. Each (red) dot represents an attack that occurred from one Src. IP. As shown in Fig. 2, attack duration is widely distributed in the log-linear scale.

3. Denied-events based detection method

In order to reduce the load of active servers from long time attacks, we propose a denied-events based detection method using events of firewall. SSH brute-force attacks sometimes try to access to inactive servers. These accesses are denied by firewall rules. We groups denied events into Src. or Dst. IPes. If the number of denied events in the clustered Src. or Dst. IP group exceed pre-defined thresholds, then SSH brute-force attacks are detected. In this paper, we called this method 'DEBD (Drop-Events Based Detection mechanism).'

Table 1 describes firewall events were collected in our supercomputing service environment. We could detect to 2,352 SSH brute-force attacks by DEBD. Fig. 3 shows monthly SSH brute-force attack IPes. The DEBD detects more SSH brute-force attacks than the [4] because the DEBD finds up to SSH scanning attacks try to connect to every IP in supercomputing service subnetworks.

There are many attacks to SSH servers among detected attacks by the DEBD. Practically we find 1,086 attacks toward servers among detected 2,352 SSH brute-force attacks by DEBD. We expect detection of attacks ahead SSH servers through the DEBD. It can reduce the load of servers as a pre-filter. Fig. 4 shows monthly ratio of pre-filtered brute-force attacks. The average ratio is more than 46% although there is a discrepancy in monthly ratio.

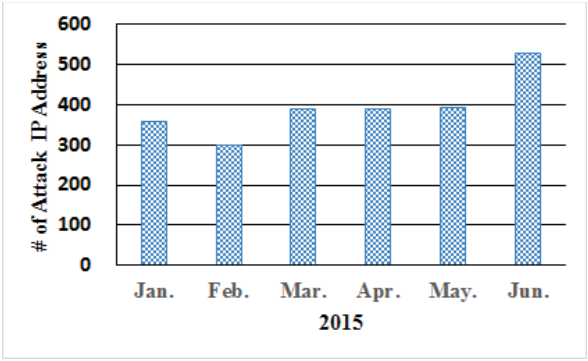


Fig. 3: Detected SSH brute-force attacks by DEBD (monthly).

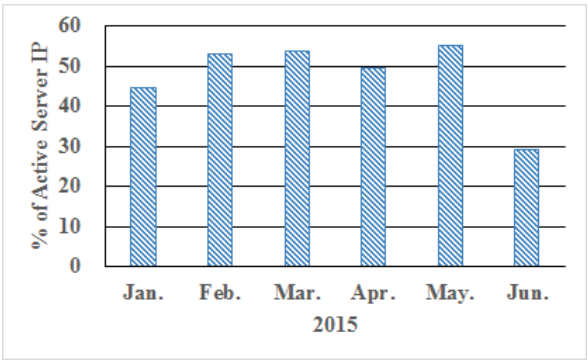


Fig. 4: Ratio of active servers IP among the detected attacks.

4. Conclusions

In this paper, we propose DEBD method against SSH brute-force attack. In our environment, the [4] detected 983 SSH brute-force attacks and the proposed DEBD detect 2,352 SSH brute-force attacks during 6 months in 2015. The proposed method filter beforehand by 46% on average that the attack traffic flow to active servers.

References

[1] T. Ylonen and C. Lonvick, The Secure Shell (SSH) Transport Layer Protocol, RFC 4253, 2006
[2] Owens, Jim, and Jeanna Matthews, "A Study of Passwords and Methods Used in Brute-Force SSH Attacks," USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008
[3] McAfee Labs Threats Report (2015). [Online]. Available:http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2015.pdf
[4] Jae-Kook Lee, Sung-Jun Kim, Joon Woo, and Chan-Yeal Park, "Analysis and Response of SSH Brute Force Attacks in Multi-user Computing Environment," KIPS Tran. on Computer and Communication Systems, Vol. 4, No. 6, Jun. 2015.
[5] Thames, J. Lane, Randal Abler, and David Keeling, "A distributed active response architecture for preventing SSH dictionary attacks," IEEE Southeast Con., 2008.
[6] A. Satoh, Y. Nakamura, and T. Ikenaga, "A flow-based detection method for stealthy dictionary attacks against Secure Shell," Journal of Information Security and Applications, 2014, Vol. 21, pp.31-41

How to realize a Soft Tracking of People in Temporary Reception Centres

Submitted As Poster

G.L. Masala[1] and M.L. Ganadu[2],
[1]Department of Political Science, Communication, Engineering and Information Technologies and
[2] Department of Humanities and Social Sciences, University of Sassari, Sassari, ITALY.

Abstract *- One of the main issues of the people authentication/tracking is related to sense of control that is poorly perceived by people. The paper describes a project for a complete identification procedure of people with multimodal biometric approach and data management in real Italian temporary reception centres for immigrants or asylum seekers. The secure authentication needs to allow the management of several services of the centres and the control of the access is differentiated for various user profile, and location, in order to protect the database from not controlled actions. Furthermore a distributed database is managed. We propose a soft tracking of people in the temporary centres, based on fingerprints features.*

Keywords: multimodal biometrics, SIFT, clinical records

1 Introduction

In temporary reception centres for immigrants several facilities are available as refectory, beds, first complimentary objects and additionally staff of the centre have to check daily the presence of registered immigrants. In particular the temporary reception centres have to satisfy requirement of the Italian Admission Plan (PAI); PAI needs a coordination of the emergency centre for immigration with the prefectures to solve the primary identification of people, the daily reports of presence and healthy, and management of all the services.
People arrive in the temporary centres often without identity documents and they typically try to refuse any approch with picture identification.
In this paper we present a soft method of people authentication to render bearable the tracking, using fingerprints recognition; this is a real system implemented and is a credible candidate of management of the PAI.

2 Main results

Our target are five temporary centres in Sardinia (Italy) where usually less than 100 people are daily present in each centre; it may occur to accommodate new immigrants, in group of about 20 new individuals in the same time, once a month.

The reception procedure requires the registration of new applicant with biometric features and general information in the local temporary centres.

The first day of arrival, for each person a face photo and fingerprints are taken. In particular the image is taken through a CCD camera while the fingerprints using the HI-SCAN PRO BIOMETRIKA scanner (FTIR, 500 dpi, 1" x 1").

In this phase generally people collaborate with operators to the registration, having some problems only with the photo identification. In previous experiences we found negative impact in front of the camera for daily tracking. For this reason the automatic face recognition is not used as normal check of daily presence in any centres. The management of daily access to refectory, complimentary pack and registration of presence is fingerprints based.

Every day immigrants are recognised by means of their fingerprints in order to access to all services: refectory, complimentary pack, healthy treatment and at same time the presence is tracked and call the roll is not more compulsory. Other duties are related to the development and management of the electronic clinical records. Management of the administrative and information recording; essential information to establish the person's identity, former employment, status of asylum seeker and family relationships with relatives and people already present in the Italian territory.

The system is composed by two main modules. The first one allows to manage the data while the second one ensures the identity of the person. It is worth noting that data stored in clinical records are sensitive and require secure access.

Having the same central management of several centres (i.e. to define free space available, general supplies and monitoring), we have to solve a problem of data management for a distributed database and shared services. Currently the services are administrated using a web application. It is possible the access to local databases having a virtual catalogue. In this way operators can manage people in the distribute database with the same masks, and working on updated database. The file are also copied in the central management centre every day to backup purpose. We have 5 locations and in particular in the temporary centre of San Pietro in Sassari, we have the management of the other centres and the direct connection with the Italian Automated Fingerprint Identification System (AFIS) database; the network is shown in figure 1.

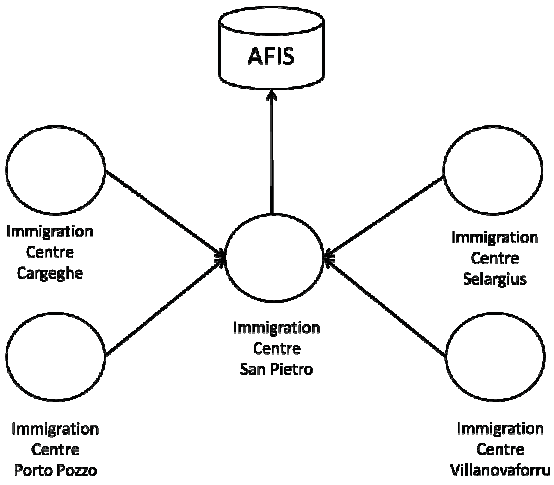


Fig. 1 Temporary reception centres in Sardinia, Italy

To facilitate learning incrementally by the user, the system allows to configure the interface according to predefined user profiles. Moreover the integration with the most widely used tool for Office Automation as MS Office® (© Microsoft Corporation Inc.).

The access to the clinical/administrative records from the operator is made by fingerprint authentication as well. Data are encrypted in the clinical records. The clinical/administrative records is web-based so it is possible to work on data from different locations. The secure channel of communication with the web server is based on a virtual private connection (VPN). Furthermore the control of the access is differentiated for various user profiles and location in order to protect the database from not controlled actions.

To recognize the identity of the person a biometric recognition is used; the relevant features are obtained using the Scale Invariant Feature Transform (SIFT) representation [1-4] either for face and for the fingerprint recognition. SIFT can identify objects even in misperception or when partially hidden, because the descriptor SIFT feature is invariant to scale, orientation and distortion affine and partially invariant to illumination changes [1]. The SIFT key points of objects are first extracted from a set of reference images and stored in a database. Our features extraction and matching algorithms are previously described in [5,6]. It is not necessary to store the fingerprints in the database but only SIFT models are recorded.

The identity module differentiates between two types of access: admission to a service or entrance to the centre. If a person is already inside the temporary centre, the use of a single fingerprint is required for the access to each service; all fingerprints are enabled and, if the check fails, an alarm system is activated. This is an efficient way to count and guarantee the service preventing multiple requests and to monitoring for the presence during the day. For the entrance to the temporary centre a multimodal authentication [5:7] is required with the associated achievement of a couple of fingerprints.

The individual biometrics are independently evaluated by the system. In each step the system takes a decision and the fusion module expresses the final response by assessing the congruence results. At the level of decision the success of all biometric tests is required. In addition this approach allows to modify the algorithm of feature extraction without changing the system.

3 Conclusions

A complete system of data management controlled by multimodal biometrics based on SIFT features to be used in Temporary Reception Centres is here presented. The system guarantees the identity of persons and facilitates data management and related services.
Current system version provides direct connection with Italian Afis database to check the immigrant fingerprint, a novel management system and backup. The aim is to use a soft daily checking of presence to render tolerable the procedure, inside normal activity as refectory and complimentary daily packs.
Explicit checkpoints or biometric-based facial recognition are considered negatively by guests of the temporary centres.

The connection between the centres, web based, suffers by scalability problems increasing users for each centre or the number of centres. A CLOUD with biometric authentication [6] solution is the next evolution of such infrastructure.

4 References

[1] Y. Ke & R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
[2] D. Lowe. Local feature view clustering for 3d object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 682–688, 2001.
[3] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60, 91–110, 2004.
[4] M. Bicego, A. Lagorio, E. Grosso & M. Tistarelli. On the use of SIFT features for face authentication. In *Computer Vision and Pattern Recognition Workshop*, IEEE Conference 35, 2006.
[5] Masala G.L.,Ganadu M.L., Multimodal Biometric Methods for Temporary Reception Centres, , on proceedings of the 2015 International Conference on Security and Management (SAM'15), Las Vegas, USA, July 27 – 30, 2015.
[6] Masala G.L., Ruiu P.,Brunetti A.,Terzo O.,Grosso E., Biometric Authentication and Data Security in Cloud Computing, on proceedings of the 2015 International Conference on Security and Management (SAM'15), Las Vegas, USA, July 27 – 30, 2015.
[7] A.K. Jain, R. Bolle & S. Pankanti (Eds.). Biometrics: personal identification in networked society. Springer Science & Business Media,1999.

SESSION

LATE BREAKING PAPERS

Chair(s)

TBA

ISBN: 1-60132-445-6, CSREA Press ©

ISBN: 1-60132-445-6, CSREA Press ©

Cyber Passport

Identity Theft Strategic Countermeasure

Cryptographic Solutions; Administrative Framework.

International Conference on Security and Management (SAM'16)

LATE BREAKING PAPER

Gideon Samid
Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, Ohio
BitMint, LLC
Gideon@BitMint.com

Abstract: Identity Theft is the fastest rising crime in the United States with about 7% of US adult population victimized annually. This frightening scope warrants a bold government intervention. Here is a detailed proposal. "Cyber Passport" addresses itself to the main threat: a breach of a merchant, bank, or government department resulting in theft of identities of millions of citizens, which for a long time live in fear of residual violations. The solution is based on two principles: (i) online transactions may require a randomized, readily replaceable, short lived code (cyber passport); (ii) the cyber passport will be comprised of a working code, and of an un-stored code which is known only to the issuing agency and to the individual recipient. When implemented these two principles will prevent a massive violation - the biggest plague today. The un-stored code cannot be stolen from any business database because it is not stored there. Hackers will still be able to steal everything but they will have to go retail, no more wholesale theft. Cyber-Passport is not a panacea, but it brings the threat down to size. The program will be optional for citizens, and voluntary for participating establishments facing the public. It will require some legislation, a non-trivial administration, and the use of modern cryptographic technology. Albeit, an organic growth implementation plan is presented herewith.

Keywords—identity theft, cyber security, cryptography.

I. INTRODUCTION

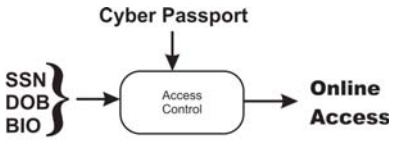
The fundamental reality that invites today' massive data theft is the situation whereby tens of thousands of public facing online establishments, (agencies), store the private financial and personal data of millions of citizens. Hackers need to find one such institution where the security is lax, and that is enough. With the data raided from a single source the hackers can inch up to the next target. The data in all these data storages is pretty much the same. And most of it is long living. Credit card companies extended the life span of their cards, so that the stolen card data is valid for many years. Of course stolen social security numbers, and stolen dates of birth are valid forever. So victims can never relax. Millions sign up with various monitoring agencies constantly on guard for emerging instances of violations. It's not just money anymore. Hackers sell personal data to people who wish to buy something unsavory online, and choose to do it via another identity. People's reputation is ruined for no fault of their own.

To counteract this situation two things are needed: deny the data thieves the opportunity to violate millions of citizens in one successful raid, and further deny them the ability to exploit their spoils for the long run. This implies that hacking will be restricted to retail data theft, and to short lived profit from such theft.

This countermeasure is all encapsulated in the Cyber Passport proposal

II. THE CYBER PASSPORT PROPOSAL

The underlying idea is to anchor identities on an off-line code, which is randomized (un-guessable), readily replaceable (quick recovery), and of two parts: one "un-stored" and the other a working code. The two parts serve as a foundation for a cryptographic protocol that is designed to (i) prevent wholesale compromise of identities, (ii) enable confidential communication that resists the Man-in-the-Middle (MiM) attack, and (iii) offers quick recovery and replacement for any compromised code. The two codes together are referred to as the Cyber Passport.



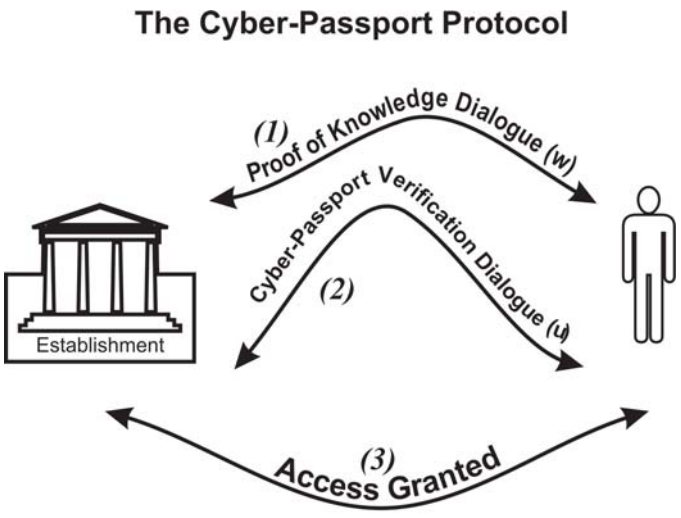
Online access granted based on life-long parameters (SSN, DoB, Bio), plus a valid short-lived, replaceable "cyber passport"

We describe ahead (i) the administration of cyber passport, and (ii) the cryptographic foundation of the same.

A. The Administration of Cyber Passport

Following the necessary legislation the government will establish a cyber passport administration which will (i) issue, maintain, and secure the cyber passport codes of the applying citizens, (ii) respond to 24/7 queries about current passports, and (iii) enforce the proper behavior of all participating institutions with public facing websites.

The detailed activities are listed ahead.



High Level Description: the government issues to applying citizens a personal, non-transferrable short code comprised, say, of 3 letters and 5 digits (all randomized), regarded as the "un-stored" code, and a second similar size code regarded as the "working code". The codes are passed off line after some formal verification of identity. The receiving citizens will use these two codes when they connect to any website where they find the icon of "participating in the cyber passport program". Such websites, per their own volition, if they are from the private sector, will apply to the government to participate in the program. Participating websites will receive the working code of their registrants, and a verifying code for the un-stored crypto passport. They will then engage the connecting citizen in a cryptographic dialogue that would convince them that he or she are in possession of their respective working code (without ever transmitting that code itself). Based on this working code the website and the connecting citizen will establish a cryptographic secure channel. Through this channel the citizen will pass his or her "un-stored" cyber passport. The participating website will not have a copy of the un-stored code, but instead will use its stored verifying code to verify that the connecting citizen is in possession of the un-stored code. Once the test is OK, the website and the citizen communicate freely using the secure cryptographic protocol they established before.

Should an individual suspect that his or her cyber passport has been compromised, he or she will apply for an instant renewed passport, which the connected websites will readily find out about in their nominal query.

Since the un-stored code is not stored by any merchant, bank, or other databases, except at the issuing authority, then by securing this one single database the government will guarantee that no wholesale theft of cyber passport will ever take place. And any retail theft, will be short lived because the codes themselves are short lived, and readily replaceable upon suspicion of theft.

1) Issue, Maintain, and Secure Cyber Passports

A dedicated administration backed by proper legislation will be established. Using non-algorithmic random number generators, the administration will issue a fresh, previously unused, number for all applicants. For example, a standard size of 4 letters, and 5 digits (easily memorable code) will cover more than 45 billion numbers, which is more than enough. And also a number that has negligible chance to be guessed.

The numbers will not be pre-stored, but generated on demand, and only then stored. There will be two codes: the 'un-stored' one and the 'working' code. The administration will work out a procedure by which each applicant is identified via off-line means, and delivered these codes also via off line means. This can happen via regular mail, or via biometric identification in a government office or via a commissioned branch of a bank, or other institution. It can be issued via states' motor vehicle administration. The recipient will then key in both codes to his communicating devices.

The administration will maintain two secure databases for the two codes. The entire proposal hinges on the premise that these two databases of all the codes can be sufficiently secured. In other words: the point in this proposal is that today's vulnerability where people's sensitive data is kept in countless databases across the country, is remedied by a situation where all this data is kept in one database center, protected by our best security people. It resolves the dilemma of the weakest link that voids the value of the high security in the other links. If there is only one link, one database center, it can be kept secure by matching our best and brightest against the best and brightest of our adversaries. Today their best and brightest match themselves against our worst and dumbest in the weakest link.

The administration will use a proper cryptographic procedure -- the blind verification procedure -- that allows one holding a verification code (but not the code itself) to verify that a communicating partner does hold the respective code. The blind verifier code will be distributed by the administration to all the participating merchants, banks, and government departments dealing with the public. The recipient establishments will keep a database of this verification code but not the un-stored code itself. A surfer trying to connect will be prompted for his cyber passport code and that code will be verified by the blind verifier code (cryptographic details ahead). The net result is that no hacker could raid this merchant or department and harvest millions of personal cyber passport codes because no merchant or department will have a database

of that code. And the cryptographic design is such that holding the verifier code does not allow one to use the proving procedure and pass as the code holder.

The working code, by contrast will be distributed to all the participating establishments for them to use in a cryptographic procedure that creates a secure communication channel with any individual who holds the same working code (and with no other). The working code will allow the establishment and the connecting individual to build a per-session secure communication channel in which the individual will pass his un-stored code, which the establishment will verify using the verifier code.

Should there be a successful raid on any cyber establishment -- the respective working codes will all be re-issued, and the gain to the raiders will be voided. Should an individual suspect that his code was compromised, he or she will apply for a fresh one.

The codes will be short lived by design, so that even if a code is stolen without detection it will have a short effective life.

2) Query Response

The administration for cyber passport will be ready 24/7 to respond to queries about the verifier code and the working code of any individual. All participating merchants will have secure communication channels with the administration to effect these queries and their responses.

A participating establishment, upon being accessed by an individual, will check its own database to see if this individual applied for a cyber passport. The establishment will not take the individual word for it. If no entry is found for that individual then the establishment will real time verify that this individual has not applied for his or her cyber passport. If it turns out that the individual by that name did apply -- then the current party is fraudster! And attempt will be made to round up the suspect.

If the verifier code does not verify the submission of the connecting individual then the establishment will query the administration to check if a fresh code was issued.

3) Participation Management

No individual will be compelled to participate, and no private establishment will be mandated to take part. Government departments for their consequential dealings with the public will participate by law. The idea being that as the program unfolds more and more individuals will opt to apply for their private cyber passport, to protect their identity, their bank account, their medical information etc. And as this happens, then banks and merchants will find it of a great disadvantage not to offer their customers this national protection, and will in turn apply to participate and abide by the rules of conduct that will come with it.

Any participating individual will have the right to opt out at any moment, and the same for any private sector establishment. Once out, no code will have to be submitted, but none of its protection will apply.

B. The Cryptographic Foundation of Cyber Passport

Cryptographically speaking we have two players: an establishment, E, (a government department, a merchant, a bank, a medical office, etc.), and an individual, I. Any establishment, E, maintains two databases one of verifier-code, v, and one for the working code, w, for all its registered individuals.

Some individual, I, connects to establishment, E, announcing its name or its registration code, and requests to do business with E. E will respond with a "non-repeat" dialogue with I. The dialogue will exchange data that is not a repeat of any previous dialogue with that individual. The dialogue will convince E that the party on the other side is in possession of its working code w, and also establish a secret shared key with which to encrypt their bilateral communication for this session only. Once this secure channel is established it will be used for I to pass to E, the un-stored code, u. Now, E has no possession of u, but only the possession of the corresponding verifier code, v. With v E will confirm that the individual across the line indeed is in possession of u. Once completed, E and I can use their per-session secure channel to do their business.

We shall now further elaborate on the mentioned procedures: proof of working code, w, using w to establish a secure communication channel, and the verification protocol, using v to verify possession of u.

1) Verification Of Working Code

The details of this procedure may be found in reference [Samid, 5/2016]. The concept is as follows: The establishment, E, selects a one time used random number, nonce, r, and sends r to the individual I. I and E both compute a number q which is a combination of the working code, w and r: $q = q(w,r)$, where the function q can be predetermined or specified ad-hoc. There is no secrecy to this function, and of course r is exposed, only w, and hence q are secret.

Looking at the binary representation of q, I will parse q to t successive distinct substrings, according to a pre-established, or ad-hoc procedure, which is not secret. I will then use a non-algorithmic random number generator (NARNG) to generate a random permutation of these t strings. This permutation q_t will be sent over from I to E.

E, on its side, will do the same for q and break it down to the same t substrings. Upon receipt of q_t, E will check if indeed q_t is a permutation of q: $q_t = T(q)$. If it is, then E will conclude that I is in possession of w, and will also infer the transposition key, K_t, that transposed q to q_t. $q_t = T(q, K_t)$. This bilateral new secret K_t will be the basis of the secret communication channel between I and E.

A hacker, H, without the possession of q will not be able to infer q from q_t because there are t! permutations, and because the division of q into the t substrings depended on the value of q, so that looking at q_t it is not clear how to divide it to t substrings in the first place. And because unlike w, q is different for every session, owing to the nonce, r, the information gleaned from previous sessions will not help H to infer w.

Illustration: An establishment E has mailed an individual I a secret PIN: w=7854. I, at some point tries to connect with E. She identifies herself, and in response E sends I a nonce: r= 2973, with instructions to compute q as the absoslute difference between w and r: q=|r-w|. Both I and E then compute q to be q=|7854-2973| = 4881. Expressed in straight binary:

q = 4881 = 1001100010001. E and I agree on subdividing q to t substrings by reading q from left to right, and incrementing the size of each successive substring thereby assuring that no two strings will be identical. The last substring may be of a size larger than +1 relative the previous substring. This will happen if the last string is identical with a previous substring. So:

q = 1-00-110-0010-001

The result is t=5. There are 5!=120 permutation of q. I then uses a non-algorithmic random number generator to identify the numbers 1-5 in a randomized order, say: K_t = 3,1,5,2,4. Accordingly, q_t will be constructed by moving the 3rd substring in q to position 1 in q_t, moving substring 1 in q to position 2 in q_t, etc.:

q_t = 110-1-001-00-0010

I will then deliver q_t = 1101001000010 to E. E does not know K_t, but it knows q (and also q_t), so E will now evaluate q_t to verify that it is a strict transposition of q. To do that E will first try to match the largest substring (#4 in q) over q_t. There are two locations where 0010 fits over q_t. E will first try the first one: 1101**0010**00010 (the bold letters denote the overlady). Then E will try to fit substring #5 in q: q_t = 1101**0010**0**001**0. This fitting attempt fails because it requires two substrings each of size 1 bit, which is not the case with q. So E will check the other fit for substring #4 in q: q_t = 1101001000**0010**. Then E will try fro fit substring #5: q_t = 1101**001**000**0010**. Then E will try to fit substring #3 in q (110): q_t = **110**1**001**000**0010**. Then E will try to fit substring #2 in q (00): q_t = **110**1**001000010**. And then substring #1 in q fits right in. E then concludes that q_t is a proper permutation of q. And furthermore, E now knows K_t = 3,1,5,2,4.

H, the hacker is not aware of q, only of q_t. He would not know how to subdivide q_t to substrings because the division was performed based on q. So H will have to try all possible combinations of dividing q to t substrings. Since the chosen procedure for dividing q to substrings depends only slightly on the contents of q (and mostly upon its size, |q|), the hacker will have a good guess of t (to satisfy 0.5t(t+1)=|q_t|). And will face t! options, each leading to a different working code, w. By selecting the identity of the nonce, r and the procedure q=q(r,w) the users determine the value of t (and the security parameter t!), which is the measure of security for the procedure that verifies w, and protects its identity.

2) *Establishing a Secure Communication Channel*

By communicating q_t to E, I has also communicated to it the randomized transposition key K_t, which was generated through white noise or through other non-algorithmic means. K_t will hence be a good per-session shared secret to be used for the secure communication channel between I and E.

I and E will agree on unit size, h, (bits per units), which will define a block size b=h*t bits, where t is the number of substrings to which q was divided. The non-algorithmic transposition key K_t will then be used to encrypt the conversation between I and E block by block. For better security, K_t will be used in a more elaborate protocol. A simple substitution cipher will prevent detection of partial order of the transposed result . The security of this solution is based on the size of the transposed list -- t. Since t is fully controlled by E and I, they can adjust it per the contents of their conversation.

The native security of transposition is super-exponential -- factorial. For a nominal t=50, the hacker will face 3.04140932 E+64 possible permutations. For t=100, the number of permutation rises to: 9.332621544 E+157.

The power of this method is that regardless of the size of the shared secret, w, the selected nonce, r, will determine the size of q, which in turn will determine the size of t. In practice this means that the two communicating parties will be in a position to decide the level of security they apply to every piece of communicated data. This is an important distinction compared to today's practice where the security is fixed by the used cipher, and is the same regardless of the sensitivity of the contents.

3) *Unstored code Verification*

Once E and I have established their secure communication line, they can use it for I to communicate the un-stored code, u, to E. E is not in possession of u, but is in possession of the u verification code, v=v(u). E will process the u value sent by I and compute its verification code, v'. If v=v' then E will conclude that I is in possession of the un-stored u value, despite the fact that E is not aware what it is.

III. IMPLEMENTATION OUTLOOK

The matured version of the Cyber Passport program is not an immediate prospect. It requires legislation, notoriously a slow process, it requires a government administration -- another tedious proposition, and it requires the expected slow organic growth. It may be a full decade until that maturity is achieved. However, the nature of this proposal is such that it can start small, and through various independent implementation sites, and then grow organically, implementation-site by implementation-site, which will then fuse into a larger implementation.

An "implementation site" is an organization of a group of establishments E_g, offering to their combined customers, or registrants the option to apply for a cyber passport per that group (E_g). The E_g will invest in a joint cyber passport administration that would issue these codes, and be ready, 24/7, to verify them. Applicants (I) for the code will pay for the service since they are its beneficiaries -- greater security. That application fee may be complemented by an investment from the group of establishments, E_g, since the establishments also benefit from this program -- offering their customers and

registrants a secure environment, which will serve as a powerful competitive edge versus others who don't offer the same. So between the application fees, and the E_g investment the cyber passport program may be clearly viable.

An implementation site may start humbly: a small number of establishments get together, with a small number of individuals (I) coming on board. But as this system operates, with almost zero burden, while more and more instances of would-be fraud cases are being prevented, because the fraudster did not show the right passport credentials, then through media outlet and dedicated advertisements, more and more establishments, on one hand, and more and more individuals, on the other hand, will apply to join the system. A positive inertia will develop. Then two or three developed implementation sites will join into a larger system, and so on.

The success of this grass root development will bubble up to the national initiative to implement the cyber passport program nation-wide.

Examples of Implementation sites: A group of online merchants, several banks, a pioneering state, medical establishments, a few federal departments. etc.

IV. Reference

Hochstein 2016 "Identity Theft - The Encyclopedia of Crime and Punishment" - Wiley Online Library

Jakobsson, 2007 "Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft" Indiana University Press.

Samid, 6/2016 "Time for a Cyber Passport" Digital Transactions Magazine June 2016

Samid, 5/2016 "T-Proof: Secure Communication via Non-Algorithmic Randomization" https://eprint.iacr.org/2016/474

Samid, 2009 "The Unending CyberWar" DGS Vitco http://www.amazon.com/Unending-Cyberwar-Gideon-Samid/dp/0963522043

Vikbladh 2016 "Identity Theft" Science 01 Apr 2016: Vol. 352, Issue 6281, pp. 46

Examination of the Possibilities of Reusing Certification Results between Different Assessment Schemes for Certification Authorities

Soshi Hamaguchi[1], **Toshiyuki Kinoshita, Ph.D.**[1], **and Satoru Tezuka, Ph.D.**[2]
[1]School of Computer Science, Tokyo University of Technology, Tokyo, Japan
[2]Graduate School of Media and Governance, Keio University, Kanagawa, Japan

Abstract – *This study identifies differences between two existing criteria and schemes (European Telecommunication Standards Institute (ETSI) and WebTrust for Certification Authorities (CAs)) for assessing CA. CAs are facing difficulties due to the fact that it is not allowed to share assessment results between different schemes. This study examines three possibilities of reusing assessment results.. First possibility is to achieve full mutual recognition between the two schemes, which is however the most difficult to be put into practice. Second possibility is to share assessment results of common set of requirements for both two criteria. Last possibility is to allow partial certification which may be applied to subcontractors of CAs such as datacenter operator.*

Keywords: PKI, Assessment, Certification, ETSI, WebTrust

1 Introduction

Information Technology is a critically important factor to support growth of the global economy, and now the technology to ensure the trust of identity on internet is a key for further development. As trust anchors of online communication, CAs are considered as vulnerable targets for attacks. In order to protect users, major browser vendors require CAs to be audited every year to be included in browsers' root stores as trusted root CA (a CA included in a browser's root store is called a "public CA"). When a user accesses a certain server which has a certificate issued by a public CA, the browser displays a message looks like Image 1. The closed-padlock image indicates that the server is trustworthy. These certificates for servers are called "SSL certificates".

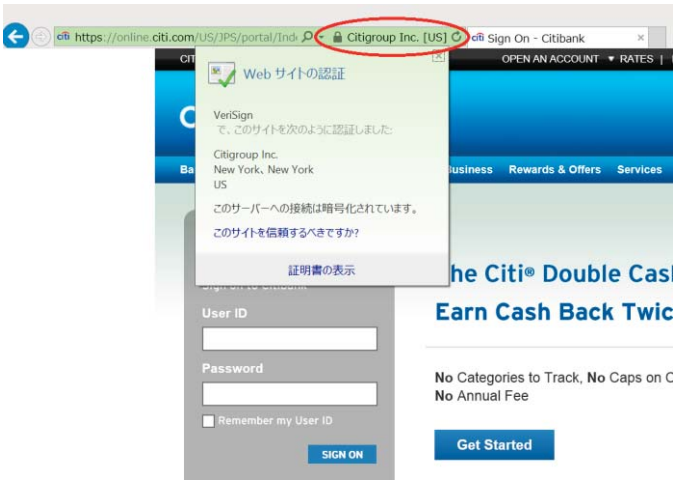


Image 1, Validated SSL Certificate on Internet Explorer

Two assessment criteria, ETSI TS 102 042 [1] and WebTrust for CAs [2], are well-known to the major browser benders and CAs because these two criteria are adopted by major browser vendors for their trusted root CA programs. A CA needs to be assessed by a trusted third party regarding fulfillment of the requirements of either ETSI or WebTrust for CA in order to claim the conformance to these criteria. The assessment burdens include not only the financial burden but also the administrative buden such as preparing documents and evidences. And due to the new EU regulation [3] that is going to require ETSI certification to the CAs established in EU member states, there are some cases where CAs are required to switch the certification. The cost for switching certification is also very high as ETSI and WebTrust for CAs have different structures, strategies, detail levels and schemes [4]. This study examines the possibilities to reduce the burden on CAs and assessors by identifying the differences and common parts between both criteria and considering the nature of CA to support trust in the digital world.

2 Related Works

Kirc Hall introduced different standards and regulations applicable to CAs in his white paper "Standards and Industry

Regulations Applicable to Certification Authorities [5]". Kirc Hall summarized in the paper that the CAs today are subjected considerable common security standards and industry regulations, imposed on CAs by the CAs themselves and by the browsers and applications as condition to be included in trusted root stores [5]. This paper introduced following standards and regulations.

- WebTrust for CAs
- ETSI audit standards
- Extended Validation Guidelines
- Baseline Requirements
- Network and Certificate System Security Guidelines
- Additional Browser Root Program Requirements

This study covers all the listed standards and regulations above except for "Additional Browser Root Program Requirements". "Additional Browser Root Program Requirements" are required by individual browser vender on top of the common set of requirements defined by Certification Authorities and Browser Forum (CAB Forum) and updated frequently. Therefore, by nature, it is very difficult to identify the similarities and to include in the scope of the study.

Thijs R. Timmerman compared the requirements in ETSI TS 102 042 and WebTrust for CAs and identified the differences in his thesis "Certificate Authority Criteria in User Perspective [4]" in 2014. His study identifies some gaps between two criteria and shortcomings of both criteria. In order to examine the possibility of reducing the burden of assessment, the identified gaps are referred in this paper.

3 ETSI standards and WebTrust for CA

Unlike WebTrust for CAs, ETSI has several relevant standards set for CAs, but this study only covers ETSI TS 102 042 as it is the comparable one to WebTrust for CAs. ETSI TS 102 042 specifies multiple Certification Policies so that CAs can chose most adequate one for them. The following is a list of certification policies specified in ETSI TS 102 042. NCP and LCP are the policies for CAs issuing end-entity certificates, and therefore only EVCP, OVCP and DVCP are relevant for this study.

•NCP (+) Normalized Certification Policy

- Equivalent to QCP except for no legal requirement as qualified trust service provider

•LCP (+)Light weight Certification Policy

•EVCP Extended Validation Certificate Policy

•OVCP Organization Validation Certification Policy

•DVCP Domain Validation Certification Policy

+: with Secure Signature Creation Device

WebTrust for CAs speficifies 3 set of criteria as follows.

•Principals and Criteria for Certification Authorities

•Principals and Criteria for Certification Authorities – Extended Validation Audit Criteria

•SSL Baseline Requirements Audit Criteria

For CAs issuing SSL certificate, all the above 3 set are relevant and for this study as well.

3.1 Assessment Scheme

Assessment and certification in accordance with ETSI standards are often performed by certification body accredited by national accreditation body. This scheme is very similar to ISO certification scheme. Figure 1 below is from ETSI TS 119 403 [6], and it shows the basic elements of the assessment scheme for ETSI. Capabilities of conformity assessment bodies are ensured by the national accreditation bodies, and the national accreditation bodies are regulated by European Co-operation for Accreditation. However, it is true that there are some assessment bodies that have no proper accreditation specified in ETSI standards.

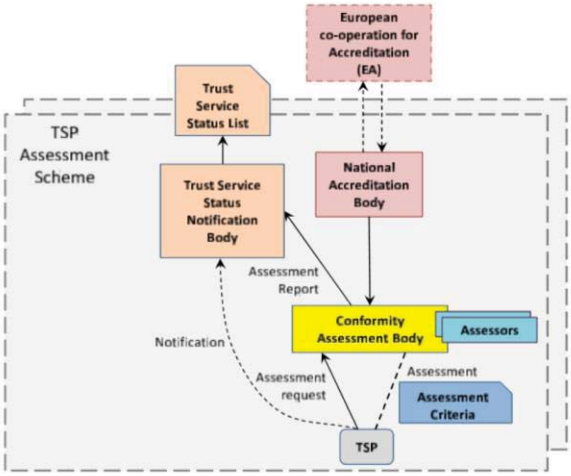


Figure 1, Assessment Scheme for ETSI [6]

On the other hand, assessments of Web Trust for CAs are performed by independent accountant firms which are recognized by AICPA/CICA. So if you compare the scheme

of Web Trust for CAs to ETSI schme, AICPA/CICA can be regarded as accredication bodys, and recognized accountant firms can be regarded as certification bodys. Two schemes look very similar, but ETSI has a higher level of organization which is European Co-operation for Accreditation. ETSI scheme can also be considered as more open scheme as all rules for the assessmenet are published. With regard to freaquency of assessment, ETSI assessment cycle is 3 years. It means full assessments are performed every 3 years, and annual serveilance asessments (kind of spot checks) are perfromed in between. This is one of the biggest gaps between ETSI and WebTrust for CAs as WebTrust for CAs requires full audit every year.

The Table 1 below describes main differences between ETSI and WebTrust for CAs from the scheme perspective.

	ETSI TS 102 042	WebTrust for CA
Scope	SSL Cert and End Entity Cert	SSL Cert
Audit Frequency	3 years cycle + annual surveillance audit (by ETSI standard)	Annually
Auditor Qualification	CWA 14172-2 ISO 17000 series EN 45000 series	Shall be licenced by AICPA, CICA

Table 1, Main differences between ETSI and WebTrust

3.2 EV Guidelines, Baseline Requirements and Network Security Requirements

A voluntary organization "CA/B Forum" that consists of major browser venders and CAs has specified following requirements for CAs to be included as trusted root CAs.

- Extended Validation Guidelines [7]
- Baseline Requirements [8]
- Network and Certificate System Security Guidelines [9]

ETSI TS 102 042, from Version 2.4.1 [1], includes requirements from the all above three requirements, baseline requirements and network security requirements. WebTrust for CAs [2] has adjusted its requirements as well. For baseline requirements, WebTrust for Certification Authority – SSL baseline Requirements Audit Criteria were specified in 2013 and in 2014 WebTrust Principles and Criteria for Certification Authorities – SSL baseline with network security were specified. For Extended Validation Guidelines, WebTrust has specific criteria named WebTrust Principles and Criteria for Certification Authorities – Extended Validation SSL.

So both criteria are properly aligned regarding the requirements specified by CAB.

Figure 2 below shows the relationship between the requirements of two criteria. As Thijs R. Timmerman mentions in his paper, ETSI and WebTrust have same domain of requirements, such as Certification Practice Statement, Key Life Cycle, Certificate Life Cycle, CA management and Operation and Miscellaneous / Soft controls. However the requirements in these domains have different detail levels.

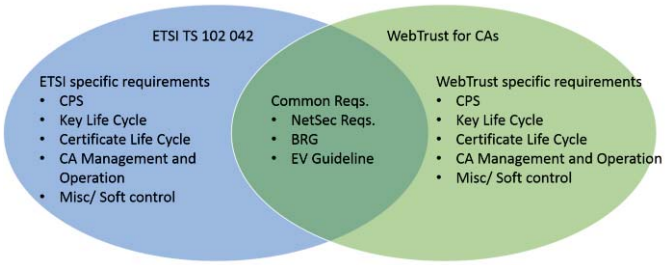


Figure 2, Common Requirements between 2 Criteria

3.3 Level of Detail

The common discussion in auditing is about the detail levels of requirements described in the criteria. If the criteria include more detailed requirements, the reproducibility of the audit result would be high. This means that the possibility for different assessors to reach the same results would be higher when more detailed requirements are included. But on the other hand, detailed requirements may deter the free interpretation of the assessor about the fulfillment of the requirements. Free interpretation allows assessors to judge not only from formal point of view but also from the back ground of the requirements. Thijs R. Timmerman has identified that WebTrust for CAs are more detailed criteria than ETSI in general [4]. However, it does not mean the ETSI is worse criteria. But this difference here about the detail levels may cause assessors to give different verdicts when assessing same CAs.

3.4 Target of Assessment / Certification

Both criteria are used to assess / certify entire CA. However, some parts of CA operations are often subcontracted to another entity. For example, not all CAs own datacenters. In this case, a datacenter facility may be shared by several CAs like figure 3 below. Figure 3 shows the case where a datacenter facility is used by three CAs and two of them are ETSI assessed / certified.

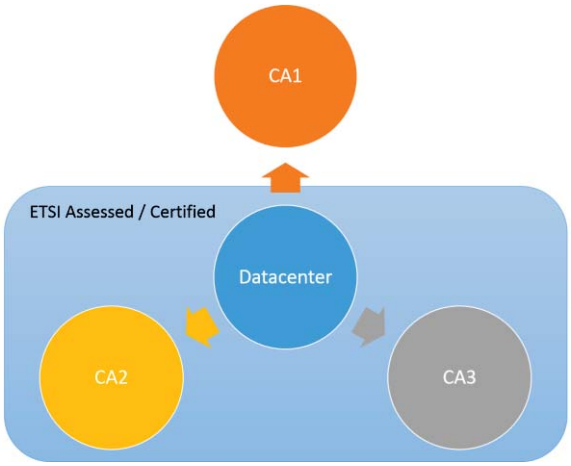


Figure 3, Shared Datacenter Facility

A question here arises, whether this datacenter has to be assessed twice a year for these two CAs. If a scheme can offer a partial certification that gives the datacenter a proof of conformance to the criteria about their operation, this certification result can be shared among these two CAs, and the CAs can claim the conformity of the datacenter facility and operation by presenting the partial certification.

4 Reuse of Certification Result

In this section, we are going to summarize the identified problems and possible options to reduce the burdens of assessment / certification. Following problems are identified in section 3.

- Difference of the Assessment Schemes (section 3.1)
- Differences of the Requirements (section 3.2 and 3.3)
- Targets of the Certification (section 3.4)

Before going into details, ETSI and WebTrust are different criteria and used in different schemes, but they have the same purpose. They have different detail levels of requirements and different nature. ETSI certification is a certification that legitimizes a body to perform certain activities in the future by assessing if the body is capable and ready to perform those activities, WebTrust is an audit that verifies whether a body has maintained a certain operational level in the past and will be able to continue maintaining the operational level [2]. Therefore it may be very difficult to achieve full mutual recognition between these two schemes. However, CA/B Forums are accepting both two schemes, and this shows the result of these two assessment / certification is equivalent for browser vendors and CAs inside CA/B Forum.

4.1 Reuse of Complete Certification

This is a possibility where ETSI Certification are recognized as an evidence of conformance to the WebTrust for CAs and vice versa. However, as described in section 4, it is very difficult to achieve full mutual recognition.

Differences between the two assessment schemes are identified in section 3.1. In order to align the quality of certification, the stricter requirements shall be respected. So the ETSI shall have full annual assessment in order to be recognized by WebTrust for CA. Another problem is the harmonization of assessment bodies. As mentioned in section 3.1, not all assessment bodies are properly accredited based on ETSI standards. Therefore, an organization should be established to publish qualification status of the assessment bodies so that it can be ensured that the assessment bodies accredited by the ETSI scheme have equivalent capabilities and quality to the assessment bodies accredited by the WebTrust scheme. Some information about the qualifications are published by ETSI(https://portal.etsi.org/TBSiteMap/ESI/TrustServiceProviders.aspx), however the information is very limited.

There is also another possibility where an assessment body has an agreement directly with WebTrust scheme. In this case WebTrust does not need to pay attention to the quality of accreditation body but to the competence and quality of the assessment body.

It should also be noted that CA/B Forums actually requires CAs to be assessed every year and this means, if a CA wants to switch the assessment scheme or to be assessed by both schemes, then the CA should pay attention to the timing of the assessments (see, following Figure 4.). It is the most effective to conduct the two assessments as close as possible so that the later assessment can utilize the result of the former assessment.

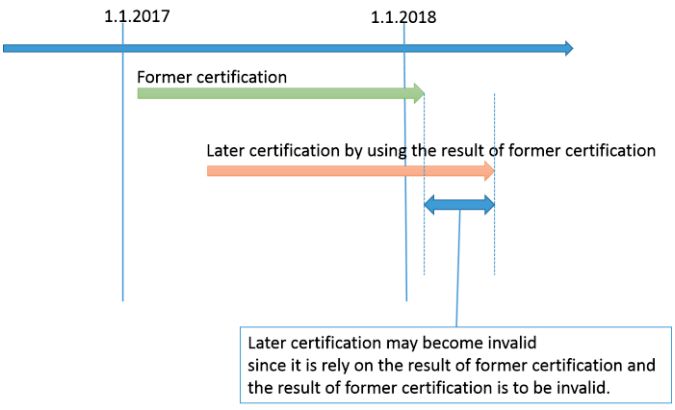


Figure 4, Validity Term of Assessment Result

4.2 Reuse of Common Requirement in the Certification

In section 3.2, the requirements that ETSI and WebTrust have in common are identified. In case CA desires to be

certified by both ETSI and WebTrust for CA certification, results of either certification can be re-used for the other certification for common set of requirements such as network security requirements and baseline requirements which are identified in section 3.2. These requirements are derived from CA/B Forums and not from ETSI or WebTrust. There should be fewer problems between two schemes. Both have equivalent detail levels. It might be more helpful to the readers of the assessment report if the results of these common requirements are described in annex or an independent report so that the reader can easily find the assessment results that should be reviewed. .

4.3 Reuse of the Partial Certification.

The idea of Partial Certification is to solve the problem about target of the assessment / certification identified in section 3.4.

Although the datacenter facilities are shared by different CAs, we identified there still some CA dependent implementation to be assessed exist such as HSMs and cabling of the computer system in the lack. These implementations are CA specific and to be checked for CA by CA. However, access control to the high secure zone, availability (power supply, cooling system and maintenances), and risk assessment against natural disasters shall be regarded common implementations for the CAs. The idea of partial certification can be applied not only between two different schemes but also within one scheme. There may be other possibilities also available as datacenter is not the only part which may be subcontracted to an entity other than CA. But the same approach described in this section can be applied.

5 Conclusions

This study examined the differences between ETSI and WebTrust from the following perspectives.

- Assessment Schemes (section 3.1)
- Differences and common requirements (section 3.2, 3.3)

Also section 3.4 identified the necessity of flexibility regarding target of assessment / certification.

This study proposes those three possible options to share assessment / certification results between two different schemes by a theoretical approach. In case the proposed possibilities are realized, CAs can reduce the cost for being assessed annually and assessment bodies can conduct faster and more cost effective assessment. This will allow all the stakeholders of this field, CAs, relying parties, browser vendors and assessment bodies, to spend more on other developments.

The establishment of harmonization bodies should be an important first step before starting negotiation between two schemes as the quality of assessment bodies for ETSI is currently difficult to check. Other ideas proposed in this study, re-use of the assessment results of common requirements and partial certification seem to be more simple and clear in theory, but mutual understanding between the two different schemes and between stakeholders are needed to proceed the discussion. We hope this study helps stakeholders to reach a consensus on the current circumstances of CA assessment.

References

[1] ETSI TS 102 042 V2.4.1, Policy requirements for certification authorities issuing public key certificates, Electronic Signatures and Infrastructures (ESI) February 2013

[2] WebTrust, WebTrust Program for Certification Authorities V2.0, AICPA/CICA, March 2011

[3] REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

[4] Certification Authority Criteria in User Perspective, Thijs R. Timmerman, August 2014

[5] Standards and Industry Regulations Applicable to Certification Authorities, Kirk Hall, Trend Micro, Inc

[6] ETSI TS 119 403 V1.1.1, Electronic Signatures and Infrastructures (ESI); Trust Service Provider Conformity Assessment - General requirements and guidance March 2012

[7] Guidelines For The Issuance And Management Of Extended Validation Certificates, V1.4, CA/Browser Forum, effective on 29 May 2012

[8] Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, V1.0, CA/Browser Forum, November 2011

[9] Network And Certificate System Security Requirements, V1.0, CA/Browser Forum, effective on 1/1/2013

Secret Sharing Scheme Based on Edge Dominating Set

Nadia M. G. Al-Saidi[1], Mohammed M. Abdulhadi[2], Mustafa Saed[3]
[1,2]Applied Sciences Department, University of Technology, Baghdad, Iraq
[3]Hyundai-Kia America Technical Center, USA
nadiamg08@gmail.com, mohammedmuthna@gmail.com, msaed@hatci.com

***Abstract*— Cryptography is the science that is closely connected with information security. While the Internet usage is constantly growing, the need for information security increases. This will necessitate new techniques to protect private information. Secret sharing scheme is one of these techniques. In this work, a new construction of a secret sharing scheme based on a special property in graph theory, edge domination, is proposed. The minimum edge dominating set is used as a minimum access structure to recover the secret key. The length of the shares (information rate ρ) in relation to the length of the secret for every given access structure is used to measure the efficiency of such schemes. Optimizing the maximum or average value for this rate is hard and constitutes a long-standing open problem in cryptography. The present work aims at improving the information rate by applying linear programming approach. Better results have been obtained when compared to those from other approaches.**

Keywords— secret sharing scheme; edge domination; information rate; access structure**.**

I. Introduction

Cryptography is the study of mathematical techniques to achieve some aspects of information security. Its main goal is to design and develop techniques and protocols to secure and authenticate data transmission and storage. Securing data storage from eavesdropping and modification, and devising counter measures to confront threats of destruction attacks in addition to problems posed by storage devices are achieved by making many copies of the secret. To facilitate this goal, new protocols were proposed in 1979 by Shamir [1] and Blakeley [2] referred to as secret sharing scheme. It is a method to split secret information S into n pieces called shares $y_1,y_2,\ldots,y_n$, such that each of them has no information about the secret S, but collecting some of them may reconstruct the secret S. Using this scheme, the secret S is shared among a set of participants $P=\{p_1,p_2,\ldots,p_n\}$ [3]. If all groups of P of at least some fixed length are qualified to obtain the secret, then this type of scheme is called threshold scheme [1]. Choosing the secret S is carried out by a special participant, called the dealer, who is also responsible of distributing the shares.

The construction of the secret is done by a collection of subsets of participants called an access structure Γ. It is monotone if for $A\in\Gamma$ and $A\subseteq B\subseteq P$ then $B\in\Gamma$. A minimal qualified subset $X\in\Gamma$ is a subset of participants, such that $Y\notin\Gamma$ for all $Y\subseteq X$, $Y\neq X$. The basis of Γ denoted by Γ_0, is the family of all minimal qualified subset. The collection of all subsets of P is given by 2^P. For any $\Gamma_0\subseteq\Gamma$, the closure of Γ_0 is defined as $cl(\Gamma_0)=\{X':\exists\ X\in\Gamma_0,\ X\subseteq X'\subseteq P\}$. Therefore, an access structure Γ is the same as the closure of its basis Γ_0, $cl(\Gamma_0)$.

A number of papers [4, 5, and 6] dealt with the design of the access structure and its relation to secret sharing scheme. An efficient construction of such a scheme was a challenge that motivates many to reach such construction, but still there is no general method. From a practical point of view, the secret sharing scheme represented by graph is very attractive. In such a scheme (based on graph theory) the vertices of the graph represent the access structure. This was referred to as graph access structure [5, 6, 7], and the idea behind the scheme was introduced by Stinson et al [8-10]. It is based on graph decomposition through finding all subgraphs of Γ in which some of them (λ) should contain all edges of Γ. This λ-decomposition leads to an increase in the number of participants (vertices) due to increasing number of sub-graphs, which gives raise to an exponential construction time.

To overcome such weaknesses, the current study proposes a new technique for the design of the access structure. In this design, the edges of the graph $E=\{e_1,e_2,\ldots,e_n\}$ are used to represent the participants $P=\{p_1,p_2,\ldots,p_n\}$. This is different from other graph theory approaches because they consider the set of vertices as participants. Besides; the minimum dominating set of edges in G will be used to represent the minimum access structure Γ_0.

In addition to this introductory section, the rest of this paper is organized as follows: some preliminary concepts related to edge dominating set are presented in Section 2. Two of the previous works on secret sharing scheme that are based on graph access structure, and the proposed method are introduced in this section 3. The use of linear programming

approach is included in section 4. The analysis of the proposed approach is discussed in section 5. Finally, the paper is concluded in section 6.

II. EDGE DOMINATING SET

The concept of domination in graphs was introduced by Berge [11] in 1958. In his book on graph theory, he discussed the "coefficient of external stability", which is nowadays known as the domination number of a graph. After that, the terms "dominating set," and "domination number" were introduced by Oystein Ore in 1962 when he published his book on graph theory [12]. In 1964, Yaglom and Yaglom [13] studied the problems of domination in graphs. A decade later, the notation $\gamma(G)$ was first used as the domination number of a graph G by Cockayne and Hedetniemi [14] in their survey paper.

Fundamental of domination was studied and organized in a book by Haynes [15]. Various types of dominations were studied and explained by several authors and more than 70 models of domination are described in the appendix of the book of Haynes, such as induced dominating set, vertex dominating set, and weakly connected dominating [16]. The concepts are almost the same in all these types.

In recent years, the problems of graph domination owned a great interest due to its important in many applications. One of these problems is the edge dominating that was presented by Mitchell and Hedetniemi [17]. A subset of edges X is called an edge dominating of G if every edge not in X is adjacent to at least one edge in X. The edge domination number $\mu(G)$ is the minimum cardinality of all edges dominating sets of G. An edge dominating set X is called an independent edge dominating set if no two edges in X are adjacent. The independent edge domination number $\mu'(G)$ is the minimum cardinality of all independent edge dominating sets of G.

Finding an edge dominating set is considered as an NP-hard problem. Hence, a lot of efforts are given toward proposing of an efficient algorithm for this purpose. In this work, an efficient polynomial time algorithm is designed to find the edge dominating sets with their numbers in the r-regular graph $G=(V,E)$, where $|V|=n$, and $|E|=m$. The process depends on the adjacent matrix of edges, such that for each edge in the graph, the set of its adjacent edges is computed. All the above concepts are presented in the following algorithm. The resulting set is then used as a minimum access structure in secret sharing scheme.

A. Edge dominating set for new access structure

Let G be a graph with a set of vertices denoted by V and a set of edges denoted by E. Let $P=\{p_1,p_2,\ldots,p_n\}$ be a set of participants and $Y=\{y_1,y_2,\ldots,y_n\}$ be a set of shares. The family of all qualified subsets used to reconstruct the secret S is denoted by Γ_0, and the family of all subsets that cannot be used to obtain any information about the secret is denoted by B_0. The set $\Gamma=\{\Gamma_0,B_0\}$ is called an access structure. An access structure is an (k,n)-threshold secret sharing scheme if it satisfies:

$$\Gamma_0 = \{B \in 2^P : k \le |B| \le n\}$$
$$B_0 = \{B \in 2^P : 0 \le |B| \le k-1\}$$

In secret sharing scheme, an access structure is complete if both $\Gamma_0 \cap B_0 = \Phi$ and $\Gamma_0 \cup B_0 = 2^P$ are true. An access structure is a monotone if it satisfies the following properties:

$$B \in \Gamma_0 \Rightarrow B' \in \Gamma_0 \text{ For all } B' \supseteq B$$
$$B \in B_0 \Rightarrow B' \in B_0 \text{ For all } B' \subseteq B$$

B. Algorithm for edge dominating set.

Input: number of vertices n, number of edges m, regularity degree r, and the adjacent matrix of edges

$$AE = a_{ij} = \begin{cases} 1 & if\ e_i\ is\ adjacent\ to\ e_j \\ 0 & otherwise \end{cases}.$$

Output: the minimum edge dominating set ϖ, The number of edge dominating sets l is computed by: $l = \left\lceil \frac{m}{(2r-1)} \right\rceil$

The algorithm relies on calling the Set_Creation function (Function Set_Creation(***s***, ***r***, ***m***)) recursively, where ***s*** :is the starting value for each loop, ***r*** : is the rank used each time this function is called, and ***m***: is the number of edges, s,r and m, are integers.

The number of loops required to calculate Γ_0 sets is equal to the rank ***r***, while, the number of sets depend on m. The function Set_Creation is designed to find all the possible combinations of number r. It is called r number of times recursively. The execution of this function is terminated when $r=l$ as shown below:

```
Function Set_Creation(s, r, m)
    If r = 1 Then
       for ii = s To m
          c = c + 1
          set_all(c, r) = ii
       end for
    else
       for ii = s To m - r + 1
          Set_Creation(ii + 1, r - 1, m)
          for jj = cr(r) + 1 To c
             set_all(jj, r) = ii
          end
          cr(r) = c
       end for
    end If
end
```

The following function is used to find the set Γ_0 in a given graph with its edge adjacency matric based on the rank. In this function two different matrices is generated, which are Set_Active and Set_True, in addition to Mat_Val that refers to the adjacency matrix of *G*. Set_active is consisted of all true result for the intersection between the elements of Set_all matrix according to the rank. Whereas, Set_true is consisted of the elements in the matrix Set_Active that satisfied Γ_0 conditions.

```
Function Set_Result()
  for jj = 1 To c
     vc = 0
     tmp = 0
    for kk = 1 To r
     for ll = 1 To r
      If Mat_Val(set_all(jj, ll), set_all(jj, kk)) = 1
        Then tmp = 1
      end If
          end for
        end for
       If tmp = 0 then  ac = ac + 1
          for ll = 1 To r
             Set_Active(ac, ll) = set_all(jj, ll)
          end for
        end If
     end for
     tmp = 0
 for jj = 1 To ac
    vc = 0
    for kk = 1 To m
     for ll = 1 To r
       If Mat_Val(set_Active(jj, ll), kk) = 1
          Then tmp = 1
       end If
     end for
      If tmp = 0 Then vc = vc + 1
      else
        tmp = 0
      end If
    end for
     If vc = r Then tc = tc + 1
      for ll = 1 To r
         set_True(tc, ll) = set_Active(jj, ll)
       end for
     end If
  end for
end
```

The above loop creates the last part of the set before exiting the function.

Example: The following example for a regular graph of order 2 with 5 vertices is used to explain how the minimum edge dominating set could be found using the proposed edge dominating algorithm.

Given a 2 – regular graph *G,* the rank *l* is calculated by using $l = \left\lceil \frac{m}{2r-1} \right\rceil$. Hence, $l = \lceil 5/3 \rceil = 2$.

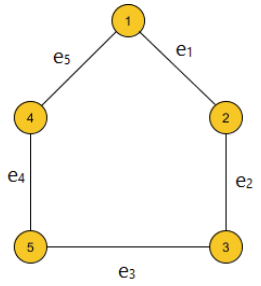


Figure 1. Regular graph of order 2 with 5 vertices

Find the adjacent matrix of edges such that;

$$AE = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The adjacent edges $A(i)$ for all edges e_i, i=1,…,5 are found as follows: when a_{ij}=1 in the row r_i, the edge e_j is adjacent (an element in $A(i)$).

$A(1) = \{e_2, e_5\}$, $A(2) = \{e_1, e_3\}$, $A(3) = \{e_2, e_4\}$
$A(4) = \{e_3, e_5\}$, $A(5) = \{e_1, e_4\}$

Now, all possible unions of any two sets are found. If the number of edges in the union sets $A(i) \cup A(j)$ is equal to *m-1*, then the edges $(e_i, e_j) \in \Gamma_0$. If we continue in the same manner, Γ_0 is constructed as follows:

$A(1) \cup A(2) = \{e_1, e_2, e_3, e_5\} \Rightarrow (e_1, e_2) \notin \Gamma_0$
$A(1) \cup A(3) = \{e_2, e_4, e_5\} \Rightarrow (e_1, e_3) \in \Gamma_0$
$A(1) \cup A(4) = \{e_2, e_3, e_5\} \Rightarrow (e_1, e_4) \in \Gamma_0$
$A(1) \cup A(5) = \{e_1, e_2, e_4, e_5\} \Rightarrow (e_1, e_5) \notin \Gamma_0$
$A(2) \cup A(4) = \{e_1, e_3, e_5\} \Rightarrow (e_2, e_4) \in \Gamma_0$
$A(2) \cup A(5) = \{e_1, e_3, e_4\} \Rightarrow (e_2, e_5) \in \Gamma_0$
$A(3) \cup A(1) = \{e_2, e_4, e_5\} \Rightarrow (e_3, e_1) \in \Gamma_0$
$A(3) \cup A(5) = \{e_1, e_2, e_4\} \Rightarrow (e_3, e_5) \in \Gamma_0$
$A(4) \cup A(1) = \{e_2, e_3, e_5\} \Rightarrow (e_4, e_1) \in \Gamma_0$
$A(4) \cup A(2) = \{e_1, e_3, e_5\} \Rightarrow (e_4, e_2) \in \Gamma_0$
$A(5) \cup A(2) = \{e_1, e_3, e_4\} \Rightarrow (e_5, e_2) \in \Gamma_0$
$A(5) \cup A(3) = \{e_1, e_2, e_4\} \Rightarrow (e_5, e_3) \in \Gamma_0$

In the construction above, *N* is computed to represent the number of possible subsets (edges) that belong to Γ_0. Hence, for this example, *N*=10. Since each set is appeared twice, then $|\Gamma_0| = 5$. Therefore, the minimum edge dominating set for this example is: $\Gamma_0 = \{(e_1, e_3), (e_1, e_4), (e_2, e_4), (e_2, e_5), (e_3, e_5)\}$.

Using the proposed algorithm, Table 1 is constructed. It tabulates the relations between the number of vertices, edges, regularity, and number of edge dominating sets. Using this table, we can

conclude a relationship between degree of regularity, rank of minimum edge dominating set and the number of edges. This relationship is formulated and proved through the following theorem that shows the improvement in the lower bound.

Theorem 3.1: Let $G(V,E)$ be an r – regular simple graph with $|E| = m$ where $m \geq 4$, if $|\varpi| = l$, then

$$lr \leq m \leq l(2r-1)$$

Proof: Let G be r – regular simple graph with $|E| = m, m \geq 4$ and $|\varpi| = l$, let $\varpi = \{e_1, e_2, \dots, e_l\}$.
Since $|\varpi| = \left\lceil \frac{m}{2r-1} \right\rceil$ and $|\varpi| = l$, then $l = \left\lceil \frac{m}{2r-1} \right\rceil$
Therefore, $l \geq \frac{m}{2r-1} \Rightarrow m \leq l(2r-1)$.

This proves the upper bound. To prove the lower bound, we need to prove $l \leq \frac{m}{r}$.
since $l = \left\lceil \frac{m}{2r-1} \right\rceil$ then we must prove $\left\lceil \frac{m}{2r-1} \right\rceil \leq \frac{m}{r}$.
Now, using mathematical induction to prove that:
$2r-1 > r$.
If $r = 2 \Rightarrow 2r-1 > r$.
If $r = k \Rightarrow 2k-1 > k$ is true.
Assume that $r = k+1$, therefore
$2(k+1)-1 > (k+1) \Rightarrow 2k+2-1 > k+1 \Rightarrow 2k+1 > k+1$ is true.
Then $\left\lceil \frac{m}{2r-1} \right\rceil \leq \frac{m}{r} \Rightarrow l \leq \frac{m}{r} \Rightarrow lr \leq m$.

III. Secret Sharing Scheme Based on Graph Access Structure

In this section, two types of secret sharing scheme based on graph access structure is presented and used in the comparison to prove the efficiency of the proposed scheme.

A. Sun et al. Scheme

Sun et al [18] introduced a new construction of a perfect secret sharing scheme of rank m. In their scheme, they assumed that $P = \{p_1, p_2, \dots, p_n\}$ is a set of participants and Γ is a uniform access structure of rank m on those participants, where Γ_0 is the basis of Γ. The decomposition of Γ_0 is Γ_i's, for $1 \leq i \leq n$ where $\Gamma_i = \{X: X \in \Gamma_0 \text{ and } p_i \in X\}$.

Thus, $\Gamma = cl(\Gamma_0) = cl(\Gamma_1) \cup \dots \cup cl(\Gamma_n)$. They then defined $\Gamma_i^* = \{X: X \cup \{p_i\} \in \Gamma_i\}$, where each $cl(\Gamma_i^*)$ is a uniform access structure of rank m-1. The secret $K = (k_1, k_2, \dots, k_m\}$, where each k_i, $1 \leq i \leq m$ is taken randomly over $GF(q^{h(m-1)})$, which is considered as the space of the secret. A polynomial $f(x)$ of degree $m.h(m-1)$-1with coefficients K is selected by a dealer to compute $y_i = f(i\text{-}1) \bmod q$, for $i = 1, \dots n.h(m-1)$. Thus, the secret can be recovered if one can get $m.h(m-1)$ or more y_i's. If one has no knowledge of any y_i, no information about the secret can be obtained. Random numbers $r_1, r_2, \dots, r_n$ are also selected by the dealer over $GF(q^{h(m-1)})$. They presumed that there exists a secret sharing scheme realizing $cl(\Gamma_i^*)$, such that, the secret is $r_i + y_i$ and the share of participant p_j is $S_j(\Gamma_i^*)$, which is given by:

$$S_i = \langle R_i, S_i(\Gamma_1^*), \dots, S_i(\Gamma_{i-1}^*), S_i(\Gamma_{i+1}^*), \dots, S_i(\Gamma_n^*) \rangle.$$

The reconstruction of the secret is done when the authorized participants collect their share together.

B. Alsaidi et al. Scheme

Kadhim et al [19,20] also represented the set of participants by the set of vertices in graph G, but they represented the minimum access structure by the minimum dominating set of vertices (*MID)*. Their scheme is summarized as follows:

The set of vertices $V = \{v_1, v_2, \dots, v_n\}$ corresponds to the set of participants $P = \{p_1, p_2, \dots, p_n\}$, and the set of all *MID* in G is corresponds to the minimum access structure Γ_0. They decomposed the graph G into n– subgraphs $G_i = (V_i, E_i)$, i=1,2,…,n, in such a way that $V_i = \{V \backslash N[v_i]\}$. The set Γ_0 is also decomposed into n $\Gamma_i s$ where $\Gamma_i = \{MID \in \Gamma_0, \ where\ p_i \in MID\}$ They defined $\Gamma_i^* = \{X: X \cup \{p_i\} \in \Gamma_i\}$.

The coefficients of the polynomial $f(x)$ are chosen randomly over $GF(q^{(m-1)!})$ and used to represent the secret $K = (k_1, k_2, \dots, k_m\}$. Hence, $f(x) = (k_1 x^{m-1} + k_2 x^{m-2} + \cdots + k_m)$. The secret K can be reconstructed by getting m or more y_i's, where y_i's are computed using $y_i = f(i) \bmod q$, $i = 1,2,\dots,n$, and the share for each participant p_i is calculated after selecting r random numbers $r_1, r_2, \dots, r_n$ by the dealer such that:

$$S_i = \langle r_i, S_i(\Gamma_1^*), \dots, S_i(\Gamma_{i-1}^*), S_i(\Gamma_{i+1}^*), \dots, S_i(\Gamma_n^*) \rangle.$$

When the authorized participants pool their share together, the secret can be reconstructed.

IV. The Proposed Secret Sharing Scheme

The construction of a perfect secret sharing scheme for the uniform access structure of rank n is proposed in this section. It is a novel construction based on the minimum edge dominating set in an r – regular graph G with m edges such that, the rank $l = |\varpi| = \left\lceil \frac{m}{2r-1} \right\rceil = n$.

The proposed secret sharing scheme consists of three phases: initialization phase, decomposition phase, and the reconstruction phase. In this work, A new decomposition of the graph is defined.

Suppose G is a graph, then G is decomposed into number of subgraphs A_i with vertices $V(A_i)$ and edges $E(A_i)$, where,

$V(A_i) = \{V \backslash u_i v_i$ are incidient on $e_i\}$.
$E(A_i) = \{E \backslash u_i v_i$ are adjacent with e_i,
for all $u_i v_i \in E(G)$, $i,j = 1,2,\dots,n\}$.

The phases of the proposed secret sharing scheme are explained below.

1. The initialization phase

Let $G=(V,E)$, be an r–regular graph. Let $P=\{e_1,e_2,\ldots e_m\}$ be the set of participants corresponding to the set E. We need to compute Γ_0 by finding all ϖ in the graph G, such that $|\varpi| = n$.

2. The decomposition phase

Decompose Γ_0 into m Γ_i's, such that $\Gamma_i = \{A \in \Gamma_0 \; where \; e_i \in A\}$. Define $\Gamma_i^* = \{X: X \cup \{e_i\} \in \Gamma_i\}$. The closure of Γ_i^* is a uniform access structure of rank n-1. It is the set of all minimum edge dominating set in the subgraph G_i. Let $K=(k_1,k_2,\ldots,k_n)$ be a secret, such that, k_i is taken randomly from $GF(q^{(n-1)!})$. A polynomial $f(x)$ of degree (n-1) is selected by the dealer, where its coefficients are $K=(k_1,k_2,\ldots,k_n)$.
Hence, $f(x) = (k_1 x^{n-1} + k_2 x^{n-2} + \cdots + k_n) \; mod \; q^{(n-1)!}$. Compute y_i's, such that, $y_i=f(i)$ mod q, i=1,2,…,m.

The dealer selects m random numbers $a_1,a_2,\ldots,a_m$ over $GF(q^{(n-1)!})$. Assume that there exists a secret sharing scheme realizing $cl(\Gamma_i^*)$ in which the secret is a_i+y_i and the share of participant e_j is $S_j(\Gamma_i^*)$. Hence, the share of participant e_i is given by $S_i = \langle a_i, S_i(\Gamma_1^*), \ldots, S_i(\Gamma_{i-1}^*), S_i(\Gamma_{i+1}^*), \ldots, S_i(\Gamma_m^*)\rangle$.

3. *The reconstruction phase*

The secret is recovered by getting n or more y_i's. Let $A=\{p_{i1},p_{i2},\ldots,p_{in}\}$ be a subset of n distinct participants such that $A \in \Gamma_0$. Hence, using Γ_0, we conclude A corresponds to one of the ϖ in G. Therefore, participant p_{i1} owns a_1, $S_1(G_2)$, ..., $S_1(G_n)$, participant p_{i2} owns a_2, $S_2(G_1)$, $S_2(G_3)$ ..., $S_2(G_n)$, ..., and participant p_{in} owns a_n, $S_n(G_1)$, $S_n(G_2)$ ...,$S_n(G_{n-1})$. Hence, a_1+y_1 can be recovered from $\{S_2(G_1), \ldots,S_n(G_1)\}$, because $\{p_{i2},\ldots,p_{in}\}$ dominates the subgraphs G_1, $a_i + y_i$ can be recovered from $\{S_1(G_i), S_2(G_i), \ldots,S_{i-1}(G_i), S_{i+1}(G_i), \ldots,S_n(G_i)\}$ and so on. This is because the participants $\{p_{i1}, \ldots,p_{ij}, \ldots, p_{in}\}$, $i \neq j$ dominates the subgraphs G_i. Thus, the participants $\{p_{i1}, p_{i2},\ldots,p_{in}\}$ can recover n of y_i's. These will recover $f(x)$ and the secret K.

V. INFORMATION RATE

In the secret sharing scheme, the ratio of the secret over the maximum length of the shares is known as the information rate:

$\rho = \log|k| \; / \; \log|S|$

The information rate ρ is used as a measure for the efficiency of a secret sharing scheme. Therefore, the main aim is to improve information rate values. There are several methods used for computing this value [8, 9, 17, 18, 20]. In this work, we used linear programming approach to compute the information rate of the secret sharing scheme that is based on edge dominating set in G. We will prove that this approach gave an improved result over other approaches for this important factor that is considered as a measure of the security of this type of protocols. The information rate can be computed for the subgraphs by applying a linear programming approach as follows:

According to the proposed decomposition, for every vertex v, define $T_v = |\{i : v \in G_i\}|$, and $T_* = max\{T_v : v \in V(G)\}$

Consider the following optimization problem:

Minimize $T_* = max\{\sum_{i=1}^{k} a_i T_{iv} : v \in V(G)\}$
Subject to : $a_i \geq 0, \; 1 \leq i \leq k$
$\sum_{i=1}^{k} a_i = 1$

where k represents the number of decomposition of graph G, and a is computed as follows: if a_i, i=1,…n appear in the subgraph G_j, s times, then $a_i=1/s$.

Therefore, the information rate $\rho = k/\min T_*$

Theorem 1. For any 2-regular graph (cyclic graph), the information rate $\rho(C_n)=2/3$, where $n \geq 4$.

Proof: To prove that $\rho(C_n)=2/3$ we should prove that $\rho(C_n) \leq 2/3$ and $\rho(C_n) \geq 2/3$.

The first part is proved using theorem 4.2 presented by Blundo [5]. Now, we have $\rho(C_n) \leq 2/3$ for connected and not complete multipartite graph. We need to show that $\rho(C_n) \geq 2/3$. To prove $\rho(C_n) \geq 2/3$, we use our proposed decomposition to decompose C_n into 2 sets as follows:

$$\{\{p_1p_n, p_1p_2, p_2p_3\}, \{p_3p_4, \ldots, p_{n-1}p_n\}\}$$

and

$$\{\{p_1p_2, p_2p_3, p_3p_4\}, \{p_1p_n, \ldots, p_{n-1}p_n\}\}$$

The maximum appearance of some of the vertices v_i, $i=1,\ldots n$ was found to be 3.
That mean max T_{pi} = 3.

Since ρ = number of decomposition/max T_{pi}
Therefore, $\rho \geq 2/3$.

VI. PERFORMANCE ANALYSIS

We have proposed an efficient construction to realize the perfect secret sharing schemes with uniform access structures based on edge dominating set. Compared with the constructions given in [9, 20], we found that, the proposed construction has some improved lower bounds for the information rate as shown in Table 2. Al-Saidi et al. [20] have shown that their information rate has an improved bound over some well-known rates found in [9,18].

Table.2: Information rate comparison

No. of edges	Stinson method in [9]	Alsaidi et al. in [20]	Proposed method

7	ρ=7/11	ρ=1/3	ρ=2/3
8	ρ=2/3	ρ=6/19	ρ=2/3
9	ρ=9/14	ρ=6/28	ρ=2/3
10	ρ=2/3	ρ=6/38	ρ=2/3
11	ρ=11/17	ρ=6/51	ρ=2/3
12	ρ=1/3	ρ=6/28	ρ=1/3
13	ρ=13/20	ρ=24/83	ρ=2/3
14	ρ=2/3	ρ=24/102	ρ=2/3
15	ρ=15/23	ρ=24/123	ρ=2/3

VII. CONCLUSION

This paper has focused on an important cryptographic primitive called secret sharing scheme. The main contribution of this paper is in proposing an efficient polynomial time algorithm to find the edge dominating set for any given graph. This set is used as an access structure in designing of perfect secret sharing scheme. The information rate for the proposed scheme has improved after using linear programing approach and satisfactory results have been obtained when the lower bound is optimized.

REFERENCES

[1] A. Shamir. "How to share a secret". Communications of the ACM, Vol. 22, No. 11, 1979, pp. 612-613.

[2] G. R. Blakley. "Safeguarding cryptographic keys". In proceedings of American Federation of Information Processing Societies 1979 National Computer Conference, Vol. 48, 1979, pp. 313-317.

[3] E.F Brickell and D.M. Davenport. "On the classification of ideal secret sharing schemes". J. Cryptology, Vol.4, 1991, pp. 123-134.

[4] A. Beimel, T. Tassa, and E. Weinreb, "Characterizing ideal weighted threshold secret sharing", SIAM J. Discrete Math. Vol. 22, 2008, pp. 360–397.

[5] C. Blundo, A. De Santis, D.R. Stinson, and U. Vaccaro, "Graph decomposition and secret sharing schemes", J. Cryptology, Vol. 8, 1995, pp. 39–64.

[6] G. Di Crescenzo and C. Galdi, ″Hypergraph decomposition and secret sharing", Discrete Appl. Math. Vol. 157, 2009, pp. 928–946.

[7] M. Ito, A. Saito, and T. Nishizeki, ″Secret sharing scheme realizing general access structure", Proc. IEEE Globecom, Vol. 87, 1987, pp. 99–102.

[8] E.F. Brickell and D.R. Stinson, "Some improved bounds on the information rate of perfect secret sharing schemes", J. Cryptology, Vol. 5 1992, pp. 153–166.

[9] D. R. Stinson. "Decomposition Constructions For Secret Sharing Schemes", IEEE Transactions information theory, Vol. IT – 40. No. 1, 1994, pp. 118 – 125.

[10] D. R. Stinson. "New general lower bounds on the information rate of secret sharing schemes". In Advance in Cryptology-CRYPTO. 92, Lecture Notes in Computer Science, Springer-Verlag, Vol. 740, 1993, pp. 168-182.

[11] C. Berge. "Theory of Graphs and its Applications" Methuen, London, 1962.

[12] O. Ore. "Theory of Graphs." Amer. Math. Soc. Colloq. Publ., 38 (Amer. Math. Soc., Providence, RI), 1962.

[13] M. Yaglom, and I. M. Yaglom. "Challenging mathematical problems with elementary solutions." Vol. 1, Combinatorial Analysis and Probability Theory, 1964.

[14] E. J. Cockayne, and S. T. Hedetniemi. "Towards a theory of domination in graphs" Networks, Vol. 7, 1977, pp. 247-261.

[15] J. McConnell. "Analysis of Algorithms: An Active Learning Approach". Jones and Bartlett publishers, 2001.

[16] D. Pierre Bovet, and P. Crescenzi. "Introduction to the theory of complexity". Prentice Hall International Series in Computer Science, 2006.

[17] S. Mitchell and S. Hedetniemi. "Edge domination in trees", Congr, Vol. 19. 1977, pp. 489 – 509.

[18] H. Sun and S. Shieh. "Constructing Perfect Secret Sharing Schemes for General And Uniform Access Structures". Journal of Information Science and Engineering, Vol. 15, 1999, pp. 679-689.

[19] K. A. Kadhim "An Improved Secret Sharing Scheme Based on Graph Theory ", MSc thesis, university of technology, 2013.

[20] N. M. Al – Saidi , N.A. Rajab, M. R. Md. Said and K.A. Kadhim. "Perfect Secret Sharing Scheme Based on Vertex Domination Set ". International Journal of Computer Mathematics. Vol. 92, No. 9, 2015.

[21] H. Sun and S. Shieh, "Recursive constructions for perfect secret sharing schemes", Comput. Math. Appl. Vol. 37, 1999, pp. 87–96.

Table 1. The relation between size, regular, and edge dominating

Ver	$reg = 2$		$reg = 3$		$reg = 4$		$reg = 5$		$reg = 6$		$reg = 7$		$reg = 8$	
	No. E	ϖ	No. E	ϖ	No. E	ϖ	No. E	ϖ	No. E	ϖ	No. E	ϖ	No. E	ϖ
4	4	2	6	2										
5	5	2	-	-	10	2								
6	6	2	9	2	12	2	15	2						
7	7	3	-	-	14	2	-	-	21	2				
8	8	3	12	3	16	3	20	3	24	3	28	3		
9	9	3	-	-	18	3	-	-	27	3	-	-	36	3
10	10	4	15	3	20	3	25	3	30	3	35	3	40	3
11	11	4	---	--	22	4	--	--	33	3	-	-	44	3
12	12	4	18	4	24	4	30	4	36	4	41	4	48	4
13	13	5	--	--	26	4	-	-	39	4	-	-	52	4
14	14	5	21	5	28	4	35	4	42	4	48	4	56	4
15	15	5	-	-	30	5	-	-	45	5	-		60	4
16	16	6	24	5	32	5	40	5	48	5	56	5	64	5
17	17	6	-	-	34	5	-	-	51	5	-	-	68	5
18	18	6	27	6	36	6	45	5	54	5	63	5	72	5
19	19	7	-	-	38	6	-	-	57	6	-	-	76	6
20	20	7	30	6	40	6	50	6	60	6	70	6	80	6

High Efficiency of Scalar Multiplication in Elliptic Curve Cryptography

Chiaki Itaba, Noboru Takeuchi, Mayuko Hirose, Kaori Katsumata
Matrazali Noorafiza, Itaru Koike, Toshiyuki Kinoshita

Graduate School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji Tokyo, 192-0982, Japan

Abstract *The elliptic curve cryptography is based on the addition and the subtraction of some points on an elliptic curve. The point obtained by repeating the addition and subtraction on the elliptic curve is called a scalar multiple point, and is used as a key of the elliptic curve cryptography. We call scalar multiplication when the formula Q=kP stands for two points P, Q on an elliptic curve and an integer k. The scalar multiplication is repeatedly used in the procedure sequence and is the most important calculation in the elliptic curve cryptography. Therefore, the speed up of the scalar multiplication can directly cause speed up of the elliptic curve cryptography. When the elliptic curve and the base point are hardly changed, the scalar multiplication can be sped up by using the pre-calculation table.*

In the this report, we propose a making algorithm of the pre-calculation table mainly using Double-Triple (DT) operation that does the double and triple multiplication at the same time.

Keywords *elliptic curve cryptography, scalar multiplication, pre-calculation table*

1. Introduction

In recent years, the performance of computers has been improved, and their prices have been cheaper. This leads to facilitate decipher and shorten the decipher time. Therefore, the problem that traditional code strength is insufficient for keeping security has been caused and the elliptic curve cryptography has been attracting attention. The elliptic curve cryptography is one of the public key cryptosystem and involves high security level since it utilizes the discrete logarithm problem that uses an elliptic curve on a finite field. Although the mathematical expression appeared in the elliptic curve cryptography is more difficult compared with other public key cryptosystems, the elliptic curve cryptography has an advantage that equivalent security level can be achieved by shorter key length. Figure 1 compares key length of the elliptic curve cryptography with that of the RSA code, that is a typical public key cryptosystem. It is said that at least 1024 bit key length is necessary to keep enough security level by the RSA code. On the other hand, it is well-known that 160 bit key length can achieve the equivalent security level by the elliptic curve cryptography. Therefore the elliptic curve cryptography is expected to be used to the built-in software with a small memory capacity.

The elliptic curve cryptography is based on the addition and the subtraction of some points on an elliptic curve. The point obtained by repeating the addition and subtraction of the same point on the elliptic curve is called a scalar multiple point, and is used as a key of the elliptic curve cryptography. Figure 2 shows the rough procedure sequence between sender and recipient of the elliptic curve cryptography. According to the figure, the scalar multiplication is repeatedly used in the procedure sequence and is the most important calculation in the elliptic curve cryptography. We call scalar multiplication when the formula $Q=kP$ stands for two points P, Q on an elliptic curve and an integer k. Therefore, the speed up of the scalar multiplication can directly cause speed up of the elliptic curve cryptography.

There are some methods for speeding up the scalar multiplication, and the Window method should be better when some room exists in the memory capacity. The Window method is a technique using the binary expansion, and drastically reduces the computational complexity of the elliptic curve cryptography. Moreover, when the elliptic curve and the

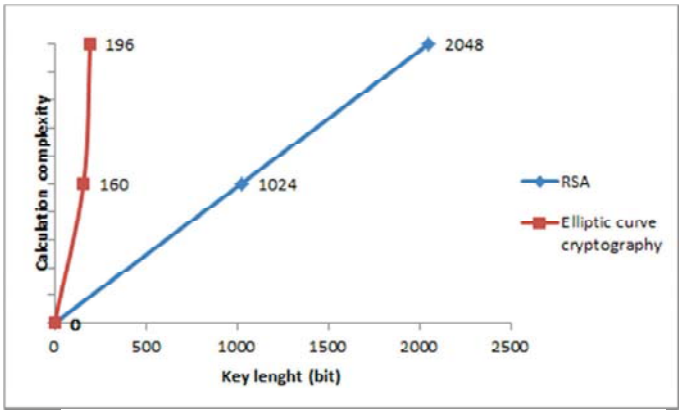


Figure 1 Comparison of the RSA encryption and elliptic curve cryptography

base point are hardly changed, the scalar multiplication can be sped up by using the pre-calculation table. Figure 3 shows a rough procedure sequence of the elliptic curve cryptography when the pre-calculation table is used.

Some methods for efficiently creating the pre-calculation table are proposed. Frac-window method is the most influential one of them that creates the pre-calculation table efficiently by using the odd number multiple points. To use this method, it is strongly important to calculate the odd number multiple points efficiently. The LM method, the LG method and the PACS method are proposed for calculating the odd number multiple points more rapidly. The LM method that uses the addition repeatedly, the LG method that use Conjugate Addition Sequence (CADD) frequently which executes the addition and the subtraction concurrently, and the PACS method that is an improvement of the LG method.

In the this report, we propose another algorithm for calculating the odd number multiple points rapidly by mainly using Double-Triple (DT) operation that calculates the double and triple multiplication at the same time.

process	Alice(sender)	Bob(recipient)
0	Share the information of the point P on the elliptic curve.	
1		Generates random number Nb as a secret key.
2		Scalar multiplication : Rb=P × Nb Set the point Rb to the public key.
3	Generates random number Na as a secret key.	
4	Scalar multiplication : Ka=Rb × Na Set the point Ka to the cryptographic key.	
5	Ciphertext C =″clear text″ M+Ka Sends C to Bob.	
6	Scalar multiplication : Ra=P × Na Set the point Ra to the public key.	
7		Scalar multiplication : Kb=Ra × Nb Set the point Kb to the cryptographic key.
	Ka=Rb × Na=P × Na × Nb Kb=Ra × Nb=P × Na × Nb Sina Ka=Kb, the key is shared.	
8		M =C−Kb Now Bob can read clear text.

Figure 2 Flow of elliptic curve cryptography

process	Alice(sender)	Bob(recipient)
0	Share the information of the point P on the elliptic curve.	
1	Creating {P,⋯,NP} pre−calculation table.	
2		Generates random number Nb as a secret key.
3		Refer to NbP form the pre−calculation table. Set the point Rb to the public key.
4	Generates random number Na as a secret key.	
5	Scalar multiplication : Ka=Rb × Na Set the point Ka to the cryptographic key.	
6	Ciphertext C =″clear text″ M+Ka Sends C to Bob.	
7	Refer to NaP form the pre−calculation table. Set the point Ra to the public key.	
8		Scalar multiplication : Kb=Ra × Nb Set the point Kb to the cryptographic key.
	Ka=Rb × Na=P × Na × Nb Kb=Ra × Nb=P × Na × Nb Sina Ka=Kb, the key is shared.	
9		M =C−Kb Now Bob can read clear text.

Figure 3 Flow of elliptic curve cryptography when using the pre-calculated table

2. Elliptic curve and Addition formula

The expression

$$y^2 = x^3 + ax + b \quad (a, b \in F_p)$$

that is defined on the prime field F_p is called an elliptic curve of the Weierstrass type, and used for the elliptic curve cryptography. The addition formula on the elliptic curve that is shown as $R = P + Q$ and is determined by the following procedures. The straight line l that passes two points P and Q on the elliptic curve crosses the curve at the third point R'. R is assumed the symmetry point of R' with respect to x axis. When the third crossing point doesn't exist, R' is assumed the point at infinity. For example, when $P(x_1, y_1)$ and $Q(x_2, y_2)$ are added, their sum $R(x_3, y_3)$ can be represented as follows.

$$r = \frac{y_2 - y_1}{x_2 - x_1}$$

$$x_3 = r^2 - x_1 - x_2$$

$$y_3 = r(x_1 - x_3) - y_1$$

From the relation R and R', it is clear that the reverse calculation of addition formula is needed in the scalar multiplication. Figure 4 shows the outline of the addition formula. Since Figure 4 is shown on an Affine plane, the intersection of the elliptic curve and the straight line that passes R' and the point at infinity creates point R.

・Affine coordinate system

In the addition formula on the Affine plane, the division operation is needed to determine the inverse element of R. As the addition formula is repeatedly executed in the elliptic curve cryptography, the number of division operations also grows large. Then, a projective coordinate system, especially Jacobian coordinate system, is usually used in the elliptic curve cryptography to avoid calculating the inverse element of R. In Jacobian coordinate system, since x and y coordinates are changed to X/Z^2, Y/Z^3 and

$y^2 = x^3 + ax + b$, $\left(\frac{Y}{Z^3}\right)^2 = \left(\frac{X}{Z^2}\right)^3 + a\left(\frac{X}{Z^2}\right) + b$ i,e,

$Y^2 = X^3 + aXZ^4 + bZ^6$ holds. Because R and R' are symmetry with respect to x axis, and the point at infinity is (1,1,0).

Jacobian coordinates involve a problem that the speed of the addition is slow although the double multiplication is fast. To solve this problem, we use Chudnovsky Jacobian coordinates that has five elements (X, Y, Z, Z^2, Z^3). In Jacobian coordinates, the computational complexity is reduced by preparing Z^2, Z^3 that are needed for addition formula.

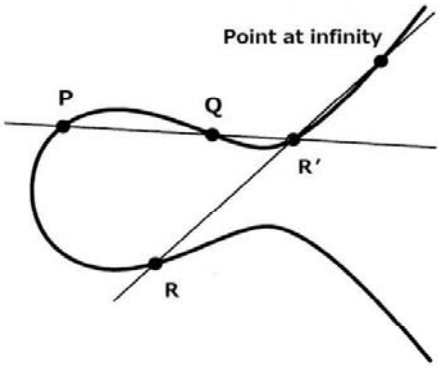


Figure 4 Additive formula

Table 1 The calculation complexity of each operation of the elliptic curve cryptography

Calculation	Calculation complexity
ADD(J)	11M+5S
ADD(Jc)	10M+4S
CADD(J)	12M+6S
CADD(Jc)	11M+5S
DBL(J)	2M+5S
DBL(Jc)	3M+8S
TPL(J)	6M+10S
TPL(Jc)	5M+7S
DT(J)	5M+7S
DT(Jc)	8M+12S

・Computational complexity

To compare the computational methods of the elliptic curve cryptography, we evaluate the computational complexity of each operation on the elliptic curve. Table 1 shows the computational complexity of each operation. The indicators M, S in this table are the computational costs of the multiplication and the square calculation on F_p respectively.

3. Previous research

One of techniques for speed up of the scalar multiplication is the pre-calculation table method, in which the results of the scalar multiplication is calculated and memorized in the table in advance. Although it takes some effort to create the pre-calculation table, the more efficient calculation can be achieved when the elliptic curve and the base point are hardly changed at every key creation.

To make the table in shorter time, the efficient calculation method is desired. Some methods of the calculation of the odd number multiplication and preservation in the pre-calculation table are introduced as follows. The arrows that indicate each calculation are explained in Figure 5.

Figure 5 Description of calculation

Figure 6 LM method

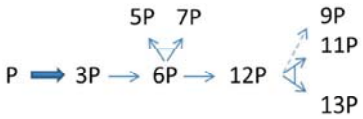


Figure 7 LG method

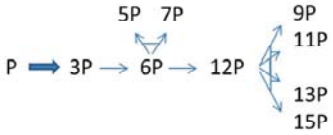


Figure 8 LG method (CADD)

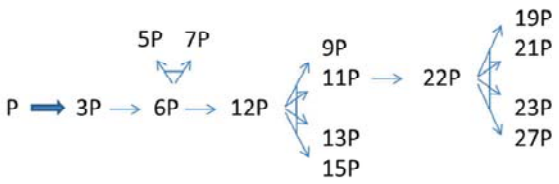


Figure 9 PCAS method

3.1 LM method

Firstly the double multiple point $2P$ is calculated, and the odd number multiple point is calculated by adding $2P$ to the base point P. Since only the addition is used in this calculation except the first calculation $2P$, the computational complexity is almost constant even if the maximum number of the table changes. Moreover, LG method has an advantage that the useless multiplication calculation does not occur after calculation of $2P$. The outline of the LM method is shown in Figure 6.

3.2 LG method

By LG method, the odd number multiple point of the pre-calculation table in the elliptic curve cryptography can be more efficiently calculated. Firstly the triple multiple point $3P$ is calculated and repeatedly calculates the double multiple point $6P$, $12P$,… . These double multiple points become strategic points to calculate the odd number multiple points. By performing CADD on the strategic point, the new odd number multiple points can be created. The points not obtained with CADD are calculated by the addition formula. When the maximum odd number multiple point of the table is 2^n-1, all the odd number multiple points can be calculated only with CADD. The computational complexity of the LG method can be less than that of the LM method in which the odd number multiple point is determined one by one, because it can determine the addition point and the subtraction point at the same time by CADD along with the maximum number of the table increasing.

Figure 7 shows the outline of the LG method. The initial point is assumed to be P, and $\{P, 3P, \ldots, 13P\}$ is calculated in the figure. Figure 8 shows that the calculation method of $\{P, 3P, \ldots, 15P\}$ when the maximum number is the 2^n-1 multiple point.

3.3 PCAS method

CADD is slightly more efficient than the addition formula. Then, improving the LG method, the technique for calculating the odd number multiple points by using only CADD not using the addition formula is Perfect Conjugate (hereafter call PCAS). In LG method and PCAS, firstly calculating the triple multiple point from base point and next double it. In PCAS, the strategic point is newly set from the odd number multiple according to the maximum odd number multiple point. From this, all the odd number multiple points can be calculated by only using CADD even when the maximum odd number multiple point of the pre-calculation table is not 2^n-1. On the other hand, it becomes the same computational complexity as the LG method according to the maximum point of the pre-calculation table.

Four functions are used to determine the strategic point from the odd number multiple point calculated by PCAS. Two functions are used properly by the table length m (odd number).

・When $m \equiv 1 \pmod 4$

1) TLP, DBL, and CADD:
   computational complexity is $23M+23S$.
2) The function that uses DBL CADD:
   computational complexity is $16M+16S$

It seems that it is efficient when judged in order of 1), 2).

・When $m \equiv 3 \pmod 4$

3) The function that uses TPL:
   computational complexity is $5M+7S$
4) DT and DBL:
   computational complexity is $8M+16S$.

It seems that it is efficient when judged in order of 3), 4).

When $m \geqq 7$, PCAS is faster than LG method that is used by the addition formula.

Figure 9 shows the outline of the PCAS. In this figure, $\{P, 3P, \ldots 27P\}$ is calculated.

4. Proposed method

In this study, the speed until complete making the pre-calculating table is emphasized, and we propose the technique for obtaining two strategic point of the double multiple point and the triple multiple point at the same time by using DT operation. In the proposed method, firstly the triple multiple point $3P$ is calculated and this point $3P$ is treated as the first strategic point similar to LM method and PCAS method. Next the double multiple point and the triple multiple point are calculated by using DT, and this double multiple point become new strategic point. After calculating the odd number multiple points, new strategic point is determined from the triple multiple point by using DT again. This is repeated until the triple multiple point reaches the table length.

Figure 10 shows the outline of the proposed method. In this figure, $\{P, 3P, 5P, \ldots, 27P\}$ is calculated.

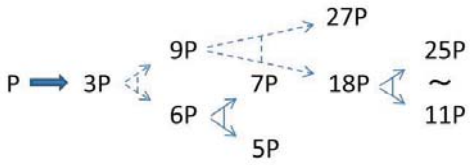


Figure 10 The proposed method

.

・The algorithm of the proposed method.

The following notations are used.
P : base point
R_1, R_2 : temporary record function
T : pre-calculation table
m : maximum odd number for requested scalar multiplication

Algorithm of proposed method

---

w1: $T \leftarrow P$
w2: $R_1 = \mathrm{TPL}(P)$
$T \leftarrow R_1$
w3: $\mathrm{DT}(R_1)$
$T \leftarrow 3R_1$
$R_1 = 2R_1$
R_2 : odd multiple point from P to less than $R_1/3$ $(=P, 3P, \ldots, (2i+1)P < R_1/3)$
w4: $T \leftarrow \mathrm{CADD}(R_1, R_2)$
If the maximum value of $T < mP$, go to w3.

---

5. Discussion

The proposed method involves a problem that when the table length m becomes larger, the calculation complexity of the existing method is larger than that of the proposed method and the gap is bigger. However, when m is smaller than or equal to 17, it is known that the calculation complexity of the proposed method is larger. The reason is because that the heavy usage of DT and the triple multiple point calculated by DT is not used. In the proposed method, when m is small, the triple multiple point determined by calculation exceeds the table length and preform useless calculation. In Figure 11, $\{P, 3P \cdots, 25P\}$ is calculated with the proposed method. Points from P to $25P$ are preserved, however $27P$ are calculated but not preserved, so this calculation is useless. Therefore, we proposed another method that uses LG method or LM method instead of DT before the calculated point exceeds the table length.

Figure 12 shows calculating $\{P, 3P \cdots, 25P\}$ by the proposed method with LG methods. As $27P$ is calculated if DT is performed on 9P, DT is replaced by LG method at this position. By this change, we can avoid that the calculated point exceeds the maximum number of the tables at the next calculation.

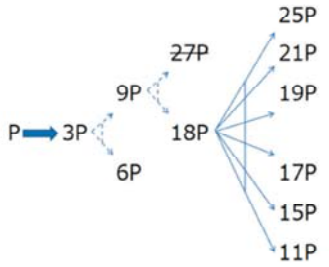


Figure 11 Problems of the proposed method

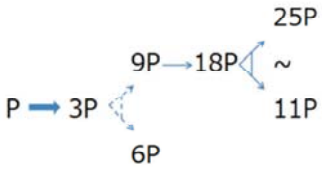


Figure 12 Mix the proposed method with LG method

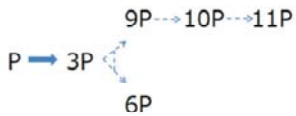


Figure 13 Mix the proposed method with LM method

・Modified proposed methods

Proposed method 2 (mix with LG method)

$m=3^n P+r$ ($3^n \leqq m < 3^{n+1}$)
w1: $T \leftarrow P$
w2: R_1= TPL(P)
$T \leftarrow R_1$
w3: DT(R_1)
$T \leftarrow 3R_1$
$R_1 = 2R_1$
R_2 : odd multiple point from P to less than $R_1/3$ (=P, $3P$, …, $(2i+1)P < R_1/3$)
w4: $T \leftarrow$CADD(R_1, R_2)
If the maximum value of $T \leqq 3^{n-1}P$, go to w3.
If $r < 2$, go to w5.
$R_1 \leftarrow$DB(R_1)
R_2 : odd multiple point from P up to less than $R_1/2$ (=P, $2P$, …, $iP < R_1/2$)
$T \leftarrow$CADD(R_1, R_2)
w5: continue

Proposed method 3 (mix with LM method)

$m=3^n P+r$ ($3^n \leqq m < 3^{n+1}$)
w1: $T \leftarrow P$
w2: R_1= TPL(P)
$T \leftarrow R_1$
w3: DT(R_1)
$T \leftarrow 3R_1$
$R_1 = 2R_1$
R_2 : odd multiple point from P to less than $R_1/3$ (=P, $3P$, …, $(2i+1)P < R_1/3$)
w4: $T \leftarrow$CADD(R_1, R_2)
If the maximum value of $T \leqq 3^{n-1}P$, go to w3.
If $r < 2$, go to w6.
w5: $R_1 \leftarrow$ADD(R_1, $2P$)
$T \leftarrow R_1$
If $R_1 < (m-2)\cdot P$, go to w5.
w6: continue

6. Future works

In the future, we are planning to clarify that in what situation the proposed method is better than the existing method, and to find the best combination with the proposed method and the existing method. In the previous research, we found that when m is between 27 and 63, mix with LG method is better than mix with LM method at m=29, 31 35, 37, 43, 47, 49, 51, 55, 57, 59, 61 and otherwise mix with LM method is better. For general m, although it cannot become the same results because the multiple points are branched from different triple point, it can be expected that mix with the LG method is better at $(m-12)P$, $(m-10)P$, $(m-6)P$, $(m-4)P$, $(m-2)P$ and mP as above mentioned, and when mix with PCAS, it is slower than usual PCAS because the strategic point judging function is not used.

Reference literature

[1] Y. Takahashi, M. Miyaji, "New pre-calculation table calculation method that uses Perfect Conjugate-Addition Sequence", IEICE Technical Report, July 2014
[2] H. Miyake, M. Miyaji, "Consideration concerning speed-up of scalar multiplication in elliptic curve cryptography", IEICE Technical Report, March 2002
[3] D. Kasahara, M. Miyaji, "Proposal of scalar multiplication by using efficient triple multiplication ", IPSJ research report, Oct 2015
[4] M. Miyaji, "Cryptology learned from algebra – mounting of elliptic curve cryptosystem from base of number theory", Nihon Hyoron Sha, March, 2012
[5] Hitachi Ltd., "Pre-calculation table making device in elliptic curve cryptography", patent official report, Feb. 2006

ISBN: 1-60132-445-6, CSREA Press ©