SECURWARE 2015

The Ninth International Conference on Emerging Security Information, Systems and Technologies

ISBN: 978-1-61208-427-5

August 23 - 28, 2015

Venice, Italy

SECURWARE 2015 Editors

Rainer Falk, Siemens AG - München, Germany

Carla Merkle Westphall, Federal University of Santa Catarina, Brazil

Hans-Joachim Hof, Munich University of Applied Sciences, Germany

SECURWARE 2015

Foreword

The Ninth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2015), held between August 23-28, 2015 in Venice, Italy, continued a series of events covering related topics on theory and practice on security, cryptography, secure protocols, trust, privacy, confidentiality, vulnerability, intrusion detection and other areas related to low enforcement, security data mining, malware models, etc.

Security, defined for ensuring protected communication among terminals and user applications across public and private networks, is the core for guaranteeing confidentiality, privacy, and data protection. Security affects business and individuals, raises the business risk, and requires a corporate and individual culture. In the open business space offered by Internet, it is a need to improve defenses against hackers, disgruntled employees, and commercial rivals. There is a required balance between the effort and resources spent on security versus security achievements. Some vulnerability can be addressed using the rule of 80:20, meaning 80% of the vulnerabilities can be addressed for 20% of the costs. Other technical aspects are related to the communication speed versus complex and time consuming cryptography/security mechanisms and protocols.

Digital Ecosystem is defined as an open decentralized information infrastructure where different networked agents, such as enterprises (especially SMEs), intermediate actors, public bodies and end users, cooperate and compete enabling the creation of new complex structures. In digital ecosystems, the actors, their products and services can be seen as different organisms and species that are able to evolve and adapt dynamically to changing market conditions.

Digital Ecosystems lie at the intersection between different disciplines and fields: industry, business, social sciences, biology, and cutting edge ICT and its application driven research. They are supported by several underlying technologies such as semantic web and ontology-based knowledge sharing, self-organizing intelligent agents, peer-to-peer overlay networks, web services-based information platforms, and recommender systems.

We take here the opportunity to warmly thank all the members of the SECURWARE 2015 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SECURWARE 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SECURWARE 2015 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SECURWARE 2015 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in emerging security information, systems and technologies.

We are convinced that the participants found the event useful and communications very open. We hope Venice provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

SECURWARE 2015 Chairs:

SECURWARE Advisory Chairs
Juha Rőning, University of Oulu, Finland
Catherine Meadows, Naval Research Laboratory - Washington DC, USA
Reijo Savola, VTT Technical Research Centre of Finland, Finland
Mariusz Jakubowski, Microsoft Research, USA
William Dougherty, Secern Consulting - Charlotte, USA
Hans-Joachim Hof, Munich University of Applied Sciences, Germany
Peter Müller, IBM Zurich Research Laboratory, Switzerland
Vladimir Stantchev, Institute of Information Systems, SRH University Berlin, Germany
Syed Naqvi, Birmingham City University, UK

SECURWARE 2015 Industry Liaison Chair
Rainer Falk, Siemens AG - München, Germany

SECURWARE 2015 Research/Industry Chair
Mariusz Jakubowski, Microsoft Research, USA

SECURWARE 2015

Committee

SECURWARE Advisory Chairs

Juha Röning, University of Oulu, Finland
Catherine Meadows, Naval Research Laboratory - Washington DC, USA
Reijo Savola, VTT Technical Research Centre of Finland, Finland
Mariusz Jakubowski, Microsoft Research, USA
William Dougherty, Secern Consulting - Charlotte, USA
Hans-Joachim Hof, Munich University of Applied Sciences, Germany
Peter Müller, IBM Zurich Research Laboratory, Switzerland
Vladimir Stantchev, Institute of Information Systems, SRH University Berlin, Germany
Syed Naqvi, Birmingham City University, UK

SECURWARE 2015 Industry Liaison Chair

Rainer Falk, Siemens AG - München, Germany

SECURWARE 2015 Research/Industry Chair

Mariusz Jakubowski, Microsoft Research, USA

SECURWARE 2015 Technical Program Committee

Habtamu Abie, Norwegian Computing Center - Oslo, Norway
Afrand Agah, West Chester University of Pennsylvania, USA
Maurizio Aiello, National Research Council of Italy - IEIIT, Italy
Jose M. Alcaraz Calero, University of the West of Scotland, United Kingdom
Alessandro Aldini, University of Urbino, Italy
Firkhan Ali Bin Hamid Ali, Universiti Tun Hussein Onn Malaysia, Malaysia
Hamada Alshaer, Khalifa University of Science, Technology & Research (KUSTAR), UAE
David Argles, Haven Consulting, UK
George Athanasiou, KTH Royal Institute of Technology, Sweden
Benjamin Aziz, University of Portsmouth, UK
Fabrizio Baiardi, University of Pisa, Italy
Ilija Basicevic, University of Novi Sad, Serbia
Lejla Batina, Radboud University Nijmegen, The Netherlands
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Francisco Jose Bellido Outeiriño, University of Cordoba, Spain
Malek Ben Salem, Accenture Technology Labs, USA
Jorge Bernal Bernabé, University of Murcia, Spain
Catalin V. Birjoveanu, "Al.I.Cuza" University of Iasi, Romania
Lorenzo Blasi, Hewlett-Packard, Italy
Carlo Blundo, Università di Salern, Italy
Wolfgang Boehmer, Technische Universitaet Darmstadt, Germany

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Risk Assessment Quantification in Hybrid Cloud Configuration

Shigeaki Tanimoto, Tsutomu Konosu, Motoi Iwashita
Faculty of Social Systems Science
Chiba Institute of Technology
Chiba, Japan
e-mail: {shigeaki.tanimoto, tklab, iwashita.motoi}@it-chiba.ac.jp

Hiroyuki Sato
Information Technology Center
The University of Tokyo
Tokyo, Japan
e-mail: schuko@satolab.itc.u-tokyo.ac.jp

Atsushi Kanai
Faculty of Science and Engineering
Hosei University
Tokyo, Japan
e-mail: yoikana@hosei.ac.jp

***Abstract*—With recent progress in Internet services and high-speed network environments, cloud computing has rapidly developed. Furthermore, the hybrid cloud configuration is now attracting attention, because it offers the advantages of both public and private clouds. However, public clouds have the problem of uncertain security, while private clouds have the problem of high cost. Thus, risk assessment in a hybrid cloud configuration is an important issue. Our previous study analyzed qualitatively risk assessment of the hybrid cloud configuration. Accordingly, through analysis of risk in a hybrid cloud configuration, 21 risk factors were extracted and evaluated, and countermeasures were proposed. However, we recognized that it was only a qualitative study and that a quantitative evaluation would be needed to make its countermeasures more practical. Hence, in this paper, the risk factors identified in the previous study are analyzed and quantitatively evaluated. Specifically, the values of the risk factors were approximately calculated by using a risk formula used in the field of information security management systems (ISMS). On the basis of these values, the effect of the countermeasures proposed in the previous study was evaluated quantitatively. It was found that the countermeasures in the previous study could reduce their corresponding risk factors by about 18% - 36%. The results herein can be used to promote hybrid cloud computing services in the future.**

Keywords-Risk Assessment; Hybrid Cloud Configuration; Risk Matrix; Risk Value Formula; Information Security Management System (ISMS)

I. INTRODUCTION

Recent years have seen great progress in Internet services and high-speed network environments. As a result, cloud computing has rapidly developed, with two main forms. First, public clouds are operated by service providers, such as Google and Amazon. Second, private clouds are built and operated by individual enterprises for their own use. Generally, a public cloud eliminates the cost of unnecessary facilities and offers rapid flexibility and scale. However, since a public cloud effectively has invisible features in a virtual configuration, enterprise users are uncertain about the cloud's security and aspects of practical use. On the other hand, a private cloud offers visualization of management, since the enterprise operates its own facilities, and guarantees security in accordance with the company's own policies. The drawbacks to a private cloud, however, include greater cost for maintenance and management of facilities, and so forth [1].

As one example, an incident of missing data and leakage by a cloud operating company, called the "big ripple," occurred in June, 2012, in Japan [2]. When a cloud provider's management handles security poorly, serious risks occur only in the public clouds that it manages, so such incidents may become apparent to users. On the other hand, a hybrid cloud form, combining aspects of both public and private clouds, is now attracting attention. Generally in a hybrid cloud, data requiring high security is handled within a private cloud, while data requiring easy operation at low cost is handled in a public cloud [3].

Thus, although the hybrid cloud form requires two different cloud forms be maintained and managed, its operation also depends on the kind of data. Furthermore, various risk factors are involved, such as accidentally saving to a different cloud during data storage [4]. For these reasons, it is important to investigate risk management in a hybrid cloud configuration. We also applied a risk assessment method for analysis and evaluation from a comprehensive viewpoint. As a result, 21 risk factors in a hybrid cloud were extracted, and countermeasures were proposed. However, it was only a qualitative study, meaning that a more practical quantitative evaluation still needed to be undertaken.

In this paper, we describe a quantitative evaluation of the risk factors of a hybrid cloud obtained in our previous study and the proposed countermeasures. Specifically, a risk value based on the formula is approximately calculated for each risk factor [5]-[7]. Then, on the basis of this value, the effect of the countermeasures on the risks can be quantitatively evaluated. It is shown that the countermeasures in the previous study can reduce their corresponding risk factors by about 18% - 36%. We believe that the results of this study will help to promote hybrid cloud computing services.

 ISBN: 978-1-61208-427-5

Section 2 reviews the hybrid cloud computing that has been studied so far. In Section 3, we describe our previous study and the present problem. Section 4 describes the quantitative evaluation of hybrid cloud computing's risks. Section 5 is a conclusion and describes future work.

II. HYBRID CLOUD CONFIGURATION

Cloud computing has now shifted to the practical use stage, and many cloud-related services increased sales in 2011. Moreover, many user companies are verifying the possibility and practicality of cloud computing in introducing information and communications technology (ICT). Cloud computing analysis is thus recognized as a key stage in systems configuration [8].

A. Reference Model of Cloud Computings

As shown in Figure 1, software as a service (SaaS), platform as a service (PaaS), and infrastructure or hardware as a service (IaaS or HaaS) are classified as the main components of the present cloud computing model. Moreover, in terms of deployment models, cloud computing is classified into public, private, and hybrid or managed clouds. Finally, cloud computing includes the roles of cloud provider and cloud user [9].

B. Hybrid Cloud

Although the hybrid cloud appears in the reference model of Figure 1, its concrete configuration combines a public cloud and private cloud, as shown in Figure 2. Usually, a company creates a hybrid cloud, and the company and a public cloud provider share executive responsibility. The hybrid cloud uses both public and private cloud services. Thus, when a company requires both public and private cloud services, a hybrid cloud is optimal. In this case, the company can summarize its service targets and service requirements and then use public or private cloud services accordingly. Thus, service correspondence can be attained in constituting a hybrid cloud, not only for a secure, mission-critical processes like employee salary processing but also for business information such as payment receipts from customers.

However, the main problem with a hybrid cloud is the difficulty of actually creating and managing such a solution. The public and private clouds must be provisioned as if they were one cloud, and implementation can become even more complicated. Therefore, since the hybrid cloud concept is a

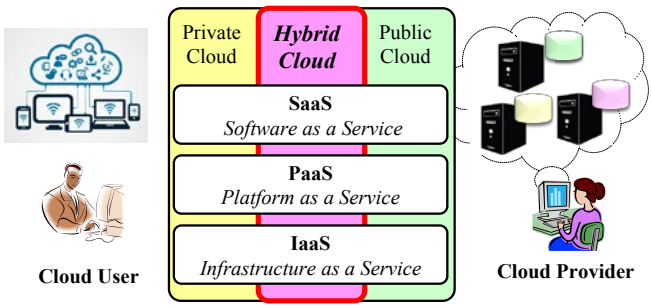


Figure 1. Reference model of cloud computing [9]

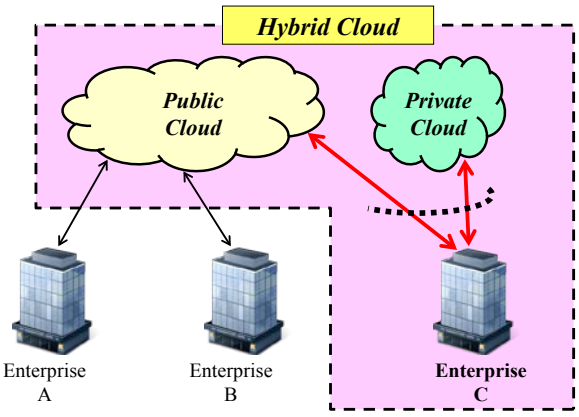


Figure 2. Hybrid cloud configuration

comparatively new architecture in cloud computing and best practices and tools have not yet been defined, companies hesitate to adopt hybrid clouds in many cases [10].

Hence, this paper examines the subject of hybrid cloud configuration in terms of these risks that adoption entails. That is, from the viewpoints of both a cloud user and a cloud provider, we consider what kinds of risks are assumed and develop a concrete risk management strategy.

C. Related work

Many security-related papers about hybrid cloud computing have been published [11]-[18]. In particular, as for the references [11] and [12], comprehensive analysis is conducted in detail about the security of cloud computing. However, the analysis of the security in a hybrid cloud configuration of these references is not sufficient. For example, these papers didn't focus on various threats in hybrid cloud computing. Also in hybrid cloud, there are threats, such as an operation mistake etc. of the cloud administrator who mentioned in Section 1. Furthermore, a user's operation mistake is also assumed as a threat peculiar to hybrid cloud. For example, when a user saves data, it is a case saved at different cloud by mistake.

On the other hand, we have already considered the risk assessment of hybrid cloud computing from the viewpoint of a user [19]. However, this evaluation is qualitative and is not sufficient. Therefore, this paper describes the risk assessment of hybrid cloud computing and adds a quantitative evaluation.

III. PREVIOUS STUDY: RISK ASSESSMENT IN HYBRID CLOUD CONFIGURATION

A. Extraction of Risk Factors

To extract the risk factors in a hybrid cloud configuration, we applied the risk breakdown structure (RBS) method, which is a typical method of risk management in project management [20]. Table 1 lists the extracted risk factors. As shown in the table, the hybrid cloud configuration was classified at the highest level into system, operation, facility, and miscellaneous categories from a comprehensive viewpoint. A total of 21 risk factors were extracted [19].

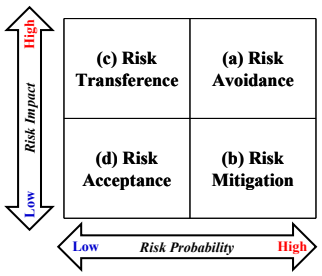
 ISBN: 978-1-61208-427-5

TABLE I. RISK FACTORS EXTRACTED BY RBS

High level	Middle level	Low level	Risk factors
1. System	1.1 Software	1.1.1 Application	1.1.1.1 A risk of mistaken allocation of the program in hybrid Cloud
			1.1.1.2 A risk of the mistaken allocation in the case of duplicate programs
		1.1.2 Data	1.1.2.1 A risk of mistaken allocation of the data in hybrid Cloud
			1.1.2.2 A risk of the mistaken allocation in the case of duplicate data
	1.2 Hardware	1.2.1 Performance	1.2.1.1 A risk of the unexpected load for CPU throughput
			1.2.1.2 A risk of unexpected use for memory size
	1.3 Network	1.3.1 Performance	1.3.1.1 A risk of the access speed slowing during network congestion etc.
2. Operation	2.1 Public cloud	2.1.1 A risk when sharing resources with the other company in public Cloud	
		2.1.2 An operation risk of public Cloud's not being administrable by the company side	
		2.1.3 A risk of the service continuity by the side of public Cloud	
	2.2 Private cloud	2.2.1 A risk of cost exceeding estimation	
		2.2.2 A risk of the human resource development in private Cloud	
	2.3 Hybrid cloud	2.3.1 A risk of the data management mismatching between different Clouds	
3. Facility	3.1 Public cloud	3.1.1 A facility risk of public Cloud's not being administrable by the company side	
		3.1.2 A risk of public Cloud's business continuity	
	3.2 Private cloud	3.2.1 A risk of an excess of facilities cost in private Cloud	
		3.2.2 A risk of the environmental construction in private Cloud	
		3.2.3 A risk of new business starting in private Cloud	
	3.3 Hybrid cloud	3.3.1 A risk of the optimal use ratio of public Cloud and private Cloud	
4. Miscellaneous	4.1 Law	4.1.1 A risk of legal revision	
	4.2 Disasters	4.2.1 A risk of a disaster	

B. *Risk Analysis in Hybrid Cloud Configuration*

Next, we devised potential countermeasures against the identified risks; these are shown in Table 2. The risk matrix method was used to deduce these countermeasures [21]. As shown in Figure 3, this method classifies countermeasures into four kinds in accordance with their risk probability and risk impact, i.e., Risk Transference, Risk Mitigation, Risk Acceptance, and Risk Avoidance. Furthermore, it gives guidelines to draw up countermeasures. Table 2 lists the classification of the risk matrix methods in correspondence with its proposed countermeasures.



Risks are classified in accordance with the risk impact and risk probability. Countermeasures corresponding to each are as follows.

(a) Risk Avoidance: A risk is avoided, and alternatives are shown.
(b) Risk Mitigation: Decrease the risk to an acceptable level.
(c) Risk Transference: Transfer a risk to a 3rd party.
(d) Risk Acceptance: Accept a risk unconditionally.

Figure 3. Risk Matrix Method

TABLE II. RISK FACTORS EXTRACTED BY RBS AND PROPOSED COUNTERMEASURES

Level 3: Risk Factors	Risk Impact	Risk Probability	Countermeasure Classification	Proposed countermeasures
1.1.1.1 A risk of mistaken allocation of the program in hybrid Cloud	High	Low	Risk transference	Strengthen the management system upon deploying data and programs.
1.1.1.2 A risk of the mistaken allocation in the case of duplicate programs	High	Low	Risk transference	Even if the cloud is used mainly on active standby, prepare an additional cloud on cold standby to enable program exchange through manual operation.
1.1.2.1 A risk of mistaken allocation of the data in hybrid Cloud	Low	High	Risk mitigation	Prepare a data management manual. Upon cloud introduction, educate and train employees.
1.1.2.2 A risk of the mistaken allocation in the case of duplicate data	High	Low	Risk transference	Even if the cloud is used mainly on active standby, prepare an additional cloud on cold standby to enable program exchange through manual operation.
1.2.1.1 A risk of the unexpected load for CPU throughput	Low	High	Risk mitigation	During cloud design, include a significant performance margin to enable efficient cloud usage even when system utilization exceeds estimates.
1.2.1.2 A risk of unexpected use for memory size	Low	High	Risk mitigation	Guarantee sufficient storage capacity to handle cases of excessive system utilization.
1.3.1.1 A risk of the access speed slowing during network congestion etc.	Low	High	Risk mitigation	During cloud design, properly consider scale, cost, enterprise usage pattern, and so forth.
2.1.1 A risk when sharing resources with the other company in public Cloud	High	High	Risk avoidance	Do not use the public cloud but protect the company by using the private cloud.
2.1.2 An operation risk of public Cloud's not being administrable by the company side	High	High	Risk avoidance	If public cloud operation is unsuitable, switch to private cloud operation, and vice versa.
2.1.3 A risk of the service continuity by the side of public Cloud	High	Low	Risk transference	Select multiple cloud providers and organize backups and other processes in other public clouds.
2.2.1 A risk of cost exceeding estimation	Low	High	Risk mitigation	Reduce cost by educating employees so that the cloud's operation can be corresponded as much as possible in its company.
2.2.2 A risk of the human resource development in private Cloud	High	Low	Risk transference	The training for the Cloud operation is held regularly in an enterprise. Accordingly, when a security incident occurs, the system which can correspond promptly is built.
2.3.1 A risk of the data management mismatching between different Clouds	Low	High	Risk mitigation	During cloud construction, fully investigate security so as to unify the security control methods of both the private and public clouds.
3.1.1 A facility risk of public Cloud's not being administrable by the company side	High	Low	Risk transference	Deploy multiple public clouds.
3.1.2 A risk of public Cloud's business continuity	High	Low	Risk transference	Take out an insurance policy upon public cloud utilization. In addition, request third-party evaluation and survey the cloud provider.
3.2.1 A risk of an excess of facilities cost in private Cloud	Low	Low	Risk acceptance	Investigate the cost of private cloud construction sufficiently, and ensure that cloud facilities are used efficiently, such as through diversion.
3.2.2 A risk of the environmental construction in private Cloud	Low	Low	Risk acceptance	If a particular situation is judged necessary for the enterprise, approve it in order to develop the business.
3.2.3 A risk of new business starting in private Cloud	Low	High	Risk mitigation	Private Cloud's operation is made to permeate as an enterprise rule beforehand.
3.3.1 A risk of the optimal use ratio of public Cloud and private Cloud	Low	High	Risk mitigation	Determine a utilization policy for data handling.
4.1.1 A risk of legal revision	Low	Low	Risk acceptance	Respond flexibly to changes in law.

 ISBN: 978-1-61208-427-5

C. Problem of the previous study

The previous study was qualitative; a more practical quantitative evaluation is needed in order to implement the countermeasures it identifies. The current study thus is a quantitative risk assessment of the risk factors obtained in our previous study and its proposed countermeasures.

IV. QUANTITATIVE EVALUATION OF HYBRID CLOUD COMPUTING'S RISKS AND PROPOSED COUNTERMEASURES

Here, the validity of a countermeasure is evaluated through a quantification of the risk factors shown in Table 2. First, a risk formula used in the field of information security management systems (ISMS) is shown [5]-[7]. Next, an approximation is described for calculating a risk value on the basis of our previous qualitative results [22]-[23]. Finally, a risk value for hybrid cloud computing services is deduced by using the formula and approximation.

A. Risk formula

Each risk value is quantified by using (1), which is used in the field of ISMS [5]-[7].

$$\textit{Risk value} = \textit{value of asset} * \textit{value of threat} * \textit{value of vulnerability} \quad (1)$$

Generally, all elements of the right-hand side of (1) are very difficult to calculate. In this paper, the following approximation is used to simplify these elements [22]-[23].

1) Approximation of the Asset Value

Here, the asset value of (1) is approximated in terms of the risk impact in the risk matrix, as shown in Figure 4. This approximation is based on the following reasons. The amount of damage was regarded for assets. As the further approximation, it was considered that the amount of damage was risk impact. Additionally, references [5]-[7] define the risk impact as 1 (low) to 5 (high). As a further approximation, these values are mapped in risk impact to a risk matrix [22]-[23]. As shown in Figure 4, the risk impact of the risk matrix is divided in two. For the sake of simplicity, the higher of the two divisions approximated to the maximum risk impact (risk value ≒ 5). Similarly, the lower of the two divisions approximated to the minimum risk impact (risk value ≒ 1).

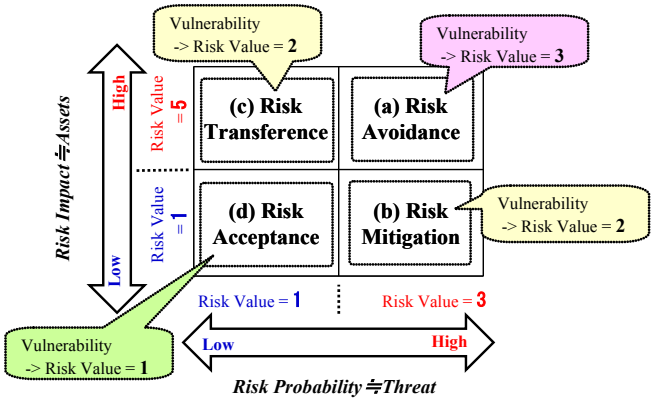


Figure 4. Risk Value Approximation of Risk Matrix [15]

2) Approximation of the Threat Value

The threat value of (1) is approximated in terms of the risk probability in the risk matrix, as shown in Figure 4. This approximation is based on the following reasons. It was supposed that threat was strongly dependent on risk probability. From references [5]-[7], the risk probability is defined to range from 1 (low) to 3 (high). These values are mapped to the generation frequencies of the risk matrix in Figure 4, as well as the above-mentioned risk impact approximation. That is, the higher of the two divisions approximated to the maximum risk probability (risk value ≒ 3), and the lower of the two divisions approximated to the minimum risk probability (risk value ≒ 1).

3) Approximation of the Value of Vulnerability

The vulnerability evaluation is defined in references [5]-[7] as well. It is defined on a three-level scale: 3 (High), 2 (Medium), and 1 (Low). These levels were approximated in accordance with the classification of the risk matrix in Figure 4. Here, the four domains of the figure are classified into three categories in accordance with the risk probability and risk impact, as follows.

- *Risk Avoidance*: both the risk probability and risk impact are high. It approximately corresponds to the highest risk classification.
- *Risk Transference and Risk Mitigation*: either the risk probability or the risk impact is high. They approximately correspond to the second highest risk classification.
- *Risk Acceptance*: both the risk probability and risk impact are low. It approximately corresponds to the lowest risk classification.

In the above-mentioned classification, *Risk Avoidance* cases are approximated to 3 (High), *Risk Transference* and *Risk Mitigation* cases to 2 (Medium), and *Risk Acceptance* cases to 1 (Low).

As mentioned above, (1) is approximated as (2). In addition, the approximate value of each parameter of (2) becomes as shown in Table 3 and Table 4.

$$\textit{Risk value} \fallingdotseq \textit{value of risk impact} * \textit{value of risk probability} * \textit{value of vulnerability} \quad (2)$$

TABLE III. APPROXIMATE VALUE OF RISK IMPACT AND RISK PROBABILITY OF (2)

	Risk Impact	Risk Probability
High	**5**	**3**
Low	**1**	**1**

TABLE IV. APPROXIMATE VALUE OF VULNERABILITY OF (2)

	Vulnerability
Risk Avoidance	**3**
Risk Transference and Risk Mitigation	**2**
Risk Acceptance	**1**

TABLE V. RISK VALUE BEFORE COUNTERMEASURES AND AFTER COUNTERMEASURES

Level 3: Risk Factors	Proposed Countermeasures	Asset ≒ *Risk Impact*	Threat ≒ Risk Probability	Vulnerability			Value of Risk		
				Before Countemeasure	After Countermeasure		Before Countermeasure	After Countermeasure	
					Ideal	Actual		Ideal	Actual
1.1.1.1 A risk of mistaken allocation of the program in hybrid Cloud	Design reinforcement of the Cloud construction	5	1	2	0	1	10	0	5
1.1.1.2 A risk of the mistaken allocation in the case of duplicate programs	Design reinforcement of the Cloud construction	5	1	2	0	1	10	0	5
1.1.2.1 A risk of mistaken allocation of the data in hybrid Cloud	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
1.1.2.2 A risk of the mistaken allocation in the case of duplicate data	Design reinforcement of the Cloud construction	5	1	2	0	1	10	0	5
1.2.1.1 A risk of the unexpected load for CPU throughput	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
1.2.1.2 A risk of unexpected use for memory size	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
1.3.1.1 A risk of the access speed slowing during network congestion etc.	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
2.1.1 A risk when sharing resources with the other company in public Cloud	Unapplied	5	3	3	3	3	45	45	45
2.1.2 An operation risk of public Cloud's not being administrable by the company side	Unapplied	5	3	3	3	3	45	45	45
2.1.3 A risk of the service continuity by the side of public Cloud	Unapplied	5	1	2	2	2	10	10	10
2.2.1 A risk of cost exceeding estimation	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
2.2.2 A risk of the human resource development in private Cloud	Unapplied	5	1	2	2	2	10	10	10
2.3.1 A risk of the data management mismatching between different Clouds	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
3.1.1 A facility risk of public Cloud's not being administrable by the company side	Unapplied	5	1	2	2	2	10	10	10
3.1.2 A risk of public Cloud's business continuity	Unapplied	5	1	2	2	2	10	10	10
3.2.1 A risk of an excess of facilities cost in private Cloud	Design reinforcement of the Cloud construction	1	1	1	0	1	1	0	1
3.2.2 A risk of the environmental construction in private Cloud	Design reinforcement of the Cloud construction	1	1	1	0	1	1	0	1
3.2.3 A risk of new business starting in private Cloud	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
3.3.1 A risk of the optimal use ratio of public Cloud and private Cloud	Design reinforcement of the Cloud construction	1	3	2	0	1	6	0	3
4.1.1 A risk of legal revision	Unapplied	1	1	1	1	1	1	1	1
4.2.1 A risk of a disaster	Unapplied	5	1	2	2	2	10	10	10
Total							221	141	182

B. *Calculation of risk value*

The risk values before applying countermeasures against risks were calculated using (2) (see Table 5 (Before Countermeasure)).

Next, the risk values after applying countermeasures were calculated. The following countermeasure was chosen from the viewpoint of practicality: "design reinforcement of the Cloud construction". This countermeasure can be easily implemented, although its costs may be problematic. Table 5 (After Countermeasure) shows the resulting risk values when performing the countermeasures.

Here, supposing an ideal case, vulnerability was assumed to be 0 as a result of using the proposed countermeasure. Moreover, supposing an actual case, this countermeasure is not always perfect. Thus, the vulnerability of an actual case is approximated to 1 (the minimum level).

C. *Results of evaluation*

Table 6 summarizes the results shown in Table 5. Although only the "design reinforcement of the Cloud construction" countermeasure was evaluated in this study, this table shows that the risk can be reduced between about 18% and 36%. These results also show that a detailed numerical expression can treat a risk more specifically by quantifying it and the prospective countermeasure.

D. *Discussion*

As mentioned above, it is not realistic to perform all of the proposed countermeasures on the risks in Table 2. Thus, this study dealt with only one ("design reinforcement of the Cloud construction") chosen on the basis of its practicality.

However, as mentioned above, the problem of cost might also affect this countermeasure. Generally speaking, this countermeasure can become expensive because it needs a specialist's knowledge. In the future, we will have to devise a verification considering such cost.

TABLE VI. EVALUATION RESULTS (SUMMARAIZATION OF RISK VALUE BEFORE COUNTERMEASURES AND AFTER COUNTERMEASURES)

	Before countermeasure against risk factors (①)	After countermeasure against risk factors (②)	
		Ideal case	Actual case
Total risk value	**221**	**141**	**182**
Risk reduction rate =(①−②)/①	-	**0.36**	**0.18**

V. CONCLUSION AND FUTURE WORK

We are interested in promoting hybrid cloud computing services as a next-generation digitized infrastructure by assessing their risks and proposing countermeasures. In our previous study, although countermeasures were developed from a qualitative risk assessment, their effectiveness could not be quantified. Hence, in this study, we performed a quantitative evaluation that used a risk value. It was shown that countermeasures labeled "design reinforcement of the Cloud construction" in the previous study could reduce their corresponding risk factors by about 18% - 36%. These results mean that the countermeasures developed in our previous qualitative evaluation can be more specifically evaluated as to their effect by introducing a risk value.

In the future, we will further improve countermeasures and verify their cost effectiveness.

ACKNOWLEDGMENTS

This work was supported by the Japan Society for the Promotion of Science (JSPS, KAKENHI Grant Number 24300029).

REFERENCES

[1] S. Nakahara, N.Fujiki, S.Ushijima, Cloud traceability (CBoC TRX), NTT technical journal, 2011.10, pp. 31-35, (In Japanese)

[2] O. Inoue, First server Failure, Nihon Keizai Shimbun, 2012.6.26, (In Japanese)

[3] Microsoft : Shift to Hybrid Cloud, (In Japanese), [Online]. Available from: http://www.microsoft.com/ja- jp/opinionleaders/economy_ict/100701_2.aspx, 2015.4.5

[4] S. Tanimoto, M. Hiramoto, M. Iwashita, H. Sato, and A. Kanai, Risk Management on the Security Problem in Cloud Computing, IEEE/ACIS CNSI 2011, Korea

[5] M. S. Toosarvandani, N. Modiri, and M. Afzali, "The Risk Assessment and Treatment Approach in order to Provide LAN Security based on ISMS Standard," International Journal in Foundations of Computer Science & Technology (IJFCST), pp. 15-36, Vol. 2, No. 6, Nov., 2012

[6] H. Sato, T.Kasamatsu, T. Tamura, and Y. Kobayashi, "Information Security Infrastructure," Kyoritsu Shuppan Co., Ltd., 2010, (in Japanese)

[7] ISMS Risk Assessment Manual v1.4, [Online]. Available from: https://www.igt.hscic.gov.uk/KnowledgeBaseNew/ISMS%20Risk%20Assessment%20Manual%20v1.4.pdf, 2015.1.4

[8] A. Goto, T.Nishihara, The concept for the cloud computing technology CboC, NTT technical journal 2009.9, pp. 64-69, (In Japanese)

[9] N. Uramoto, Security and Compliance Issues in Cloud Computing, IPSJ Magazine 50(11), 1099-1105, 2009-11-15

[10] D. Amrhei, (In Japanese), http://www.ibm.com/developerworks/jp/websphere/techjournal/0904_amrhein/0904_amrhein.html

[11] S. Sengupta, V. Kaulgud, and V. S. Sharma, Cloud Computing Security - Trends and Research Directions, 2011 IEEE World Congress on Service, pp. 524-531, 2011

[12] S.Subashini, V.Kavitha, A Survey on security issues in service delivery models of cloud computing, Elsevier, Journal of Network and Computer Applications 34 (2011)1-11

[13] K. Y. Oktay, et al., Risk-Aware Workload Distribution in Hybrid Clouds, 2012 IEEE Fifth International Conference on Cloud Computing, pp. 229-236, 2012

[14] S. Nepal, C. Friedrich, L. Henry, and S. Chen, A Secure Storage Service in the Hybrid Cloud, 2011 Fourth IEEE International Conference on Utility and Cloud Computing, pp. 334-335, 2011

[15] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, Cloud Computing Security: From Single to Multi-Clouds, 2012 45th Hawaii International Conference on System Sciences, pp. 5490-5499, 2012

[16] M. K. Srinivasan, K Sarukesi, P. Rodrigues, S. Manoj M, and Revathy P, State-of-the-art Cloud Computing Security Taxonomies - A classification of security challenges in the present cloud computing environment, ICACCI-2012, pp. 470-476

[17] L. Savu, Cloud Computing -Deployment models, delivery models, risks and research challenges-, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5778816

[18] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, Security SLAs for Federated Cloud Services, 2011 Sixth International Conference on Availability, Reliability and Security, pp. 202-209, 2011

[19] S. Tanimoto, et al., A Study of Risk Management in Hybrid Cloud Configuration, Springer, Computer and Information Science, vol. 493, pp. 247-257, 2013

[20] Risk Breakdown Structure, [Online]. Available from: http://www.justgetpmp.com/2011/12/risk-breakdown-structure-rbs.html, 2014.12.30

[21] Cox's risk matrix theorem and its implications for project risk management, [Online]. Available from: http://eight2late.wordpress.com/2009/07/01/cox%E2%80%99s-risk-matrix-theorem-and-its-implications-for-project-risk-management/, 2014.12.30

[22] S. Tanimoto, et al., A Study of Risk Assessment Quantification in Cloud Computing, 8th International Workshop on Advanced Distributed and Parallel Network Applications (ADPNA-2014), pp. 426-431, 2014

[23] S. Tanimoto, H. Sato, and A. Kanai, Risk Assessment Quantification of Ambient Service, ICDS 2015 : The Ninth International Conference on Digital Society, pp.70-75, Lisbon, Feb., 2015

 ISBN: 978-1-61208-427-5

Network Security Incident Detection Based on Network Topology Patterns

Juris Viksna, Karlis Freivalds,
Mikus Grasmanis and Peteris Rucevskis

Institute of Mathematics and
Computer Science
Riga, Latvia
Email: {juris.viksna,karlis.freivalds,
mikus.grasmanis,peteris.rucevskis}@lumii.lv

Baiba Kaskina and Varis Teivans

CERT.LV, Institute of Mathematics
and Computer Science
Riga, Latvia
Email: {baiba.kaskina,
varis.teivans}@cert.lv

***Abstract*—In this work, we explore the option of using graph topology patterns for security incident detection in NetFlow data. NetFlow data sets in which data flows related to attacks are specially marked are analyzed using graph visualization techniques in combination with manual methods to identify prospective network topology patterns related to attacks. These patterns are subsequently validated and their merit for incident detection assessed. The current research shows that while such pattern based approach is unlikely to provide a highly reliable incident detection method on its own, it can well complement other methods and can detect attacks that remain unnoticed by statistical analysis of network traffic.**

Keywords*—*Network security; Data visualization; Graph topology patterns.

I. INTRODUCTION

Network traffic data that is widely available for security incident detection in real time mostly is limited to information contained in NetFlow data format. There are several types of attacks (DoS, port scanning) that often can be detected by simple statistical analysis of NetFlow traffic and its changes over time, however such statistical analysis alone has limited capabilities for security incident detection. The possibilities to extract more information from NetFlow data have been extensively studied and one of the most widely used approaches involves use of different techniques of data visualization in combination with pattern identification in visualized data (a comprehensive survey of such approaches is presented for example in [1]). The usefulness of such methods is additionally demonstrated by commercial success of a number of proposed approaches of such type, e.g., *NFlowVis* system [2].

At the same time from the published use-cases, it is often not very clear what the capabilities and limitations of such methods are. In particular, few attempts seem to be devoted to formalization of patterns in visualized data that might indicate the presence of security incidents. Formal definition of such patterns is also unlikely to be achieved without formalization of what is exactly meant by data visualization. In this aspect, the most promising for formal treatment appear to be graph-based traffic visualization methods, not least because graphs themselves can be described by simple and well known mathematical structures, providing potential for formal definition of patterns of graph topology changes that might be indicative for specific types of attacks. For comparatively small graph topology patterns there is also a good prospect for development of efficient algorithms for detection of patterns in real time.

In graph based-representations of network traffic most often vertices represent traffic sources and edges network connections between them, however more complicated assignments, in particular for edges, are also possible. Probably one of the most formalized treatment of graph-based network monitoring is presented in [3], where a number of different graph patterns and security incidents associated with them have been identified. The analysis however is mainly done in terms of statistical attributes of such patterns (graph connectivity, average vertex degrees, etc.) and not their topology. Another comparatively formal treatment of graph patterns is presented in [4], but here the authors are focusing on the problem of network load monitoring and not on incident detection.

In our work, we use graph-based network traffic visualization with the aim to try to formally define patterns of graph *topology* that can be strongly associated with security incidents. For the study we use labeled data sets of NetFlow data integrated with data from application log files or data obtained by Deep Packet inspection (similar types of labeled data sets have been used and analyzed in [5]). Such labeled data sets for specific types of attacks allow to mark with high certainty the part of traffic involved in these attacks. Then a number of prospective topology patterns for specific attack type is selected and subsequently validated, including validation on sets of NetFlow data alone. The current results suggest that while certain types of attacks often have quite specific graph topology patterns, the same topology patterns are very likely to occur also in 'normal' traffic, and using them alone for security incident detection will give too many false positives. However, the situation significantly improves when these topology patterns are complemented by edge or vertex labels, derived from attributes of NetFlow records (port numbers, flags, number of flows, etc.) and it turns out to be possible to use such labeled topology patterns to detect several types of attacks with high certainty (i.e., with low number of false positives). Our approach of using labeled topology patterns somewhat resembles the one used in the study of

 ISBN: 978-1-61208-427-5

social networks [6]. However, we define patterns in a more formal way; also topological structures of social and network traffic graphs are very different.

In Section 2 of this paper we briefly describe the mathematical formalism used and some experimental results. In Section 3 the plans for future research are briefly outlined.

II. Formalization of Topology Patterns and Experimental Results

For the study, we use two types of data sets. The first data set is obtained by collecting NetFlow traffic from a number of dedicated servers which additionally provides labeling of 'bad traffic' on the basis of information from log files, together with traffic from the whole sub-network of these servers (together approximately 50 traffic nodes). Additionally, since the our institution is also one of the main internet service providers in Latvia, we have access to large amount of NetFlow data. However, this is largely unlabeled data, with only small part being manually analyzed by CERT.LV. The integrated analysis of these two types of data sets gets somewhat more complicated by the fact that the characteristic traffic patterns for internet service providers and end-user networks are different. For visualization and analysis purposes we use *Diagram Editor Engine Kit* – a powerful in-house developed graph visualization and clustering software suite.

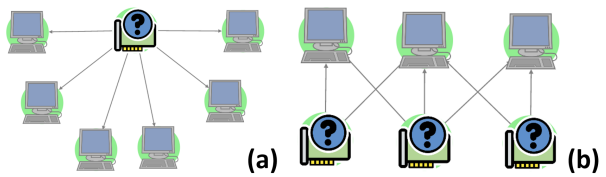


Figure 1. Two simple classes of graph topology patterns: (a) star-like patterns; (b) bipartite patterns.

The graph patterns are defined as (usually small) graphs with vertices and edges labeled by discrete and/or continuous attributes. Also each vertex and each edge is labeled by a Boolean expression having arguments in form $A \sim X$, where A is the value of attribute assigned to the particular vertex or edge, X is variable for the corresponding vertex or edge attribute in traffic graph and relation $\sim$ stands either for equality or inequality (for attributes with continuous set of values). The pattern is matched by network traffic graph, if it is found as its subgraph (or, if specified by pattern, as induced its subgraph) and all the Boolean expressions are true when their variables are substituted with attribute values from the mapped vertices or edges. Potentially perspective patterns are detected by visual analysis of network traffic graphs, and, if good candidates are found, mathematically formalized versions of these patterns are developed and then validated.

The topology of network traffic graphs being not too complex, it is not surprising that there are few patterns that are indicative of attacks by their topology alone. However, it turns out that the predictive power of such patterns considerably improves if they are complemented by edge or vertex labels derived from attributes available in NetFlow records, in which case even 'star-like' patterns that are ubiquitous in network traffic can be successfully used for detection of port scanning and DoS attacks. Examples of some simpler patterns are shown in Figure 1: star-like patterns with topological features being little specific to attacks, however with appropriate labeling added such patterns can become quite informative for attack detection; and bipartite patterns, the topology of these is already much more indicative of attacks, additional labeling is used to distinguish between different attack types.

III. Conclusions and Future Work

Our preliminary work shows that it is possible to define formal (and thus automatically detectable by a program) graph topology patterns that allow to detect security attacks with high certainty. Not all types of attacks could be linked with topology patterns however, and also the number of false negatives is comparatively high. Nevertheless such pattern-based method can detect attacks that remain unnoticed by statistical analysis of behavior of individual network nodes. Thus, they can provide a good complement to traffic statistical analysis and other widely used incident detection methods. The current aim of our research is to develop an annotated library of labeled graph topology patterns that have proved useful in incident detection together with efficient algorithms for detection of these patterns in NetFlow data.

A longer term challenge would be inclusion in pattern definitions the changes of network traffic over time. The problem of characterization of dynamic of networks is well known, however also very challenging, with very few formal results obtained. One of the most promising methods that may have some potential also for analysis of network traffic graphs is based on construction of ordered graphs of topology patterns describing their evolution with time [7].

IV. Acknowledgments

The research was supported by the project ERAF 2013/003/2DP/2.1.1.1/13/APIA/VIAA/027.

References

[1] H. Shiravi, A. Shiravi, and A. Ghorbani, "A Survey of Visualization Systems for Network Security," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, pp. 1313–1329, 2012.

[2] F. Fischer, F. Mansmann, D. Keim, S. Pietzko, and M. Waldvogel, "Large-scale Network Monitoring for Analysis of Attacks," in *Proc. of VizSec 2008*, ser. LNCS, vol. 5210, 2008, pp. 111–118.

[3] M. Iliofotou, M. Mitzenmacher, P. Pappu, S. Singh, M. Faloutsus, and G. Varghese, "Network Monitoring using Traffic Dispersion Graphs (TDGs)," in *Proc. of IMC'07*, 2007, pp. 111–118.

[4] E. Glatz, S. Mavromatidis, B. Ager, and X. Dimitropoulos, "Visualizing Big Network Traffic Data using Frequent Pattern Mining and Hypergraphs," *Computing*, vol. 96, pp. 27–38, 2014.

[5] A. Sperotto, R. Sadre, F. Vliet, and A. Pras, "A Labeled Data Set for Flow-Based Intrusion Detection," in *Proc. of IPOM 2007*, ser. LNCS, vol. 5843, 2009, pp. 39–50.

[6] R. Rossi and N. Ahmed, "Role Discovery in Networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, pp. 1112–1131, 2014.

[7] C. Vehlow, F. Beck, P. Auwarter, and D. Weiskopf, "Visualizing the Evolution of Communities in Dynamic Graphs," *Computer Graphics Forum*, vol. 34, pp. 277–288, 2015.

 ISBN: 978-1-61208-427-5

Traffic Management and Access Control in Space Experiment "Kontur-2"

Vladimir Muliukha, Vladimir Zaborovsky, Alexander Ilyashenko, Alexander Silinenko
Peter the Great St.Petersburg Polytechnic University,
Russian State Scientific Center for Robotics and Technical Cybernetics
Saint-Petersburg, Russia
e-mail: vladimir@mail.neva.ru, vlad@neva.ru, ilyashenko.alex@gmail.com, avs@rtc.ru

***Abstract*—Space experiment "Kontur-2" aboard the International Space Station (ISS) is focused on the transfer of information between station and on-ground robot. Station's resources are limited, including communication ones. That is why for the space experiment "Kontur-2" it was decided to use the methods of priority traffic management. New access control mechanisms based on these methods are researched. The usage of the priority traffic processing methods allows using more efficiently the bandwidth of receiving and transmitting equipment onboard the ISS through the application of randomized push-out mechanism. The paper considers methods of dynamic traffic management and access control that are used during international space experiment "Kontur-2" performed aboard the ISS.**

Keywords-space experiment; access control; traffic management; virtual connection

I. INTRODUCTION

Access control to the network resources is an important task of the information security. Especially it is necessary for the advanced modern space applications, for example during space experiments onboard the International Space Station (ISS). Such digital resources have to be available for authorized usage by cosmonauts and the mission control center, and protected against unauthorized access. In the modern computer networks, including ISS onboard network and satellite communicational channels TCP/IP stack is used. That is why the informational interaction between nodes is occurred using application protocols over virtual transport connections.

As the result, the problem of traffic management and access control can be presented as the task of identifying the characteristics of virtual connections and traffic control using virtual connection content code. This code is calculated according to the connection content and it shows the requested quality of service (QoS). The complexity of this problem is the fact that the content code can be calculated exactly only after the virtual connection is finished. However, in this case, the access control problem cannot be solved, because the access becomes irreversible.

The paper considers methods of dynamic priority traffic management and access control methods that have been used in international space experiments performed aboard the ISS. The proposed methods are probabilistic, but they could improve the effectiveness of information traffic management by various control throughput mechanisms of such virtual connections.

To solve the problem of calculating the dynamic content code we consider to use the indicator function, whose properties depend on: the information model of the network resource and the description of the access policy, which defines the rights of users and QoS requested by virtual connection (VC).

In this paper, we propose a new approach to access control flexibility enhancement based on active queuing management mechanism and randomized preemptive procedure. In "Kontur-2" space experiment, the offered solution was implemented by the access gateway – specialized device, based on hardware firewall that realizes several functions:

1) organization of informational interaction between robotic devices;
2) communication with operator;
3) traffic management;
4) enforce security policy.

The access gateway is a two-component device. The first part is the firewall, and the second one is the security server, which generates access rules and enforces the access policy. The adaptability of the proposed mechanism improves network security, but it requires large computational resources of the access gateway. That is why the security server was realized using cloud technologies. Security server analyzes network traffic and generates access rules for firewall considering current state of connections, available resources of internal network, and access policy.

The parameters of access gateway (firewall) rules depend on the set of network environment and/or protocols characteristics A. This set can be divided on two classes with different access conditions. In proposed approach the classification decision is based on access code F and firewall has three modes in accordance to possible F (A) values (Figure 1):

- "–1", if the data flow is forbidden according to the access policy (filtering rules);
- "1" and "0" for permitted VCs.

The state of the virtual connection is controlled throughout its lifetime. When the network environment is congested or when VCs have different QoS requirements the subset of permitted connection has to be divided into new subsets with different access codes:

- "1" for prior VCs that have low throughput and demand low stable delivery time;
- "0" for background ones that demand high throughput and have no delivery time requirements.

 ISBN: 978-1-61208-427-5

For more accurate data sorting we propose to use multiple priority levels. In our space experiment, we consider the simplest situation with two priority levels (control commands and video data). But it may be not enough for some practice tasks so we propose some easy ways to increase the number of levels using subsets of permitted VCs (see Figure 1).

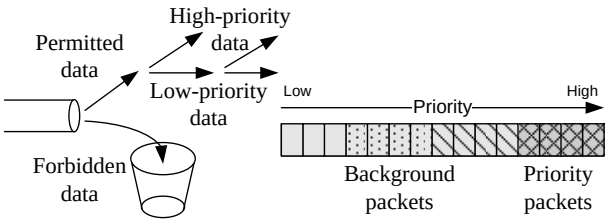


Figure 1. Multiple priority level in congested operation networks

To provide this classification procedure we have proposed active queuing management mechanism, which based on randomized preemptive control [1][2]. Therefore in the firewall, the data flow throughput and time that packets spend in queue (minimum value for priority permitted flows and infinity for denied) are the functions of randomized control parameter α.

In this paper, we propose to use the Access Gateway with the architecture described in [1]. This architecture includes several modules: Network Monitor, Access Policy Description Module, Information Resource Module, Firewall Rules Generator and Firewall that implements the rules and manages the traffic flows.

The paper is organized as follows: Section II describes the priority queueing model and basic equations. Section III is about practical application of proposed method in space experiment "Kontur-2". Section IV concludes the paper.

II. MODEL OF NETWORK ENVIRONMENT

According to the VC models written above we consider the preemptive priority queueing system with two types of customers. First type of customers has priority over the second one. The customers of the type 1 (2) arrive into the buffer according to the Poisson process with rate λ_1 (λ_2). The service time has the exponential distribution with the same rate μ for each type. The service times are independent of the arrival processes. The buffer has a finite size k $(1<k<\infty)$ and it is shared by both types of customers. The absolute priority in service is given to the customers of the first type. Unlike typical priority queueing considered system is supplied by the randomized push-out mechanism that helps precisely and accurate to manage customers of both types. If the buffer is full, a new coming customer of the first type can push out of the buffer a customer of type 2 with the probability α.

The summarized entering stream will be the elementary with intensity $\lambda=\lambda_1+\lambda_2$. Using the Kendel notation modified by Basharin, proposed system has $\vec{M}_2/M/1/k/f_2^1$ type [3].

Problems of research priority queueing have arisen in telecommunication with the analysis of real disciplines of scheduling in operating computers. During the last years, a similar sort of queueing model, and also their various generalisations are widely used at the theoretical analysis of Internet systems.

As shown in [3], the probability pushing out mechanism is more convenient and effective in comparison with other mathematical models of pushing out considered in the literature. It adequately describes real processes of the network traffic and is simple enough from the mathematical point of view. The randomized push-out mechanism helps precisely traffic management and security [4]. The other control and security factor is the telematics device buffer size. It can be varied to increase the throughput of necessary connections and reduce throughput of suspicious ones.

The state graph of system $\vec{M}_2/M/1/k/f_2^1$ is presented in Figure 2.

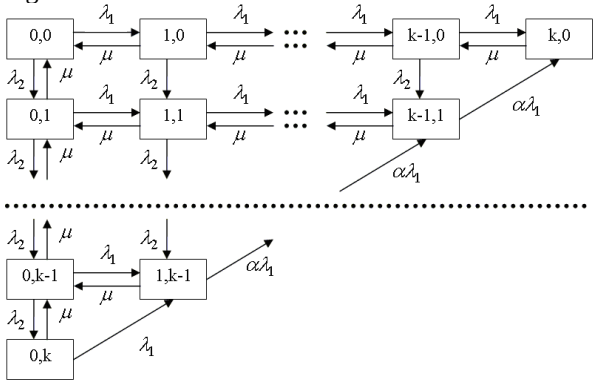


Figure 2. The state graph of $\vec{M}_2/M/1/k/f_2^1$ type system

Making by usual Kolmogorov's rules set of equations with the help of state graph we will receive:

$$\begin{aligned}&-[\lambda_1(1-\delta_{j,k-i})+\alpha\lambda_1(1-\delta_{i,k})\delta_{j,k-i}+(1-\alpha)\lambda_1\delta_{i,0}\delta_{j,k-i}+\\&+\lambda_2(1-\delta_{j,k-i})+\mu(1-\delta_{i,0}\delta_{j,0})]P_{i,j}+\mu P_{i+1,j}+\mu\delta_{i,0}P_{i,j+1}+\\&+\lambda_2P_{i,j-1}+\lambda_1P_{i-1,j}+\alpha\lambda_1\delta_{j,k-i}P_{i-1,j+1}+\\&+(1-\alpha)\lambda_1\delta_{j,k-i}\delta_{i,1}P_{i-1,j+1}=0,(0\le i\le k;0\le j\le k-i),\end{aligned}\quad(1)$$

where $\delta_{i,j}$ is the Kroneker's delta-symbol.

There is a normalization condition for the system:

$$\sum_{i=0}^{k}\sum_{j=0}^{k-i}P_{ij}=1\cdot$$

At real k (big enough) this system is ill-conditioned, and its numerical solution leads to the big computing errors. We used the method of generating functions [2][3]. According to generating function method and normalization condition we have:

 ISBN: 978-1-61208-427-5

$$G(u,v) = \sum_{i=0}^{k}\sum_{j=0}^{k-i} P_{i,j} u^i v^j, \; G(1,1) = \sum_{i=0}^{k}\sum_{j=0}^{k-i} P_{i,j} = 1$$

And after several transformations [1]-[3], solving (1) system we receive loss probability for priority ($P_{loss}^{(1)}$) and non-priority ($P_{loss}^{(2)}$) packets:

$$P_{loss}^{(1)} = q_k + (1-\alpha)\sum_{i=1}^{k-1} p_i, \qquad (2)$$

$$P_{loss}^{(2)} = r_k + \alpha\frac{\rho_1}{\rho_2}\sum_{i=1}^{k-1} p_i + \frac{\rho_1}{\rho_2} p_k \qquad (3)$$

Exploring these formulas we found some useful properties of this system described in this article. When incoming stream of priority packets getting more intensive, system starts to prohibit admission of non-priority packets. While the total flow rate is less than unity ($\rho_1 + \rho_2 \le 1$), the probability of loss is equal to zero. This means that the system is fully copes with the load.

In Figure 3 an expected result can be seen that the probability of losing priority packet decreases with increasing size of a buffer. Probability of loss is not decreasing more than 5% for small values of α . Therefore, only for large probability values increasing buffer size effectively influences the losses. For priority stream influence of this effect is the same for all values of alpha, but for non-priority packets the situation is different. Figure 6 shows that it is sometimes advantageous to have a buffer of smaller size. With a small buffer probability of be pushed out much lower, what explains this effect.

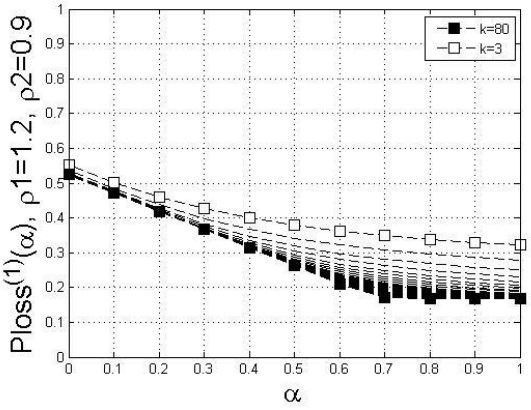


Figure 3. Loss probability of priority packets with buffer size K=3-80

Graphs of the relative throughput which is computed by formulas (4) are very important for research of processes in computer networks.

$$æ_i = 1 - P_{loss}^{(i)}, \; (i = \overline{1,2}). \qquad (4)$$

From formulas (2) and (3) we can see that by choosing parameter α , we can change $P_{loss}^{(2)}$ in very wide range. For some ρ_1 values variable $æ_i$ changes from 0.7 to 1 while $\lambda_1 + \lambda_2 >> \mu$.

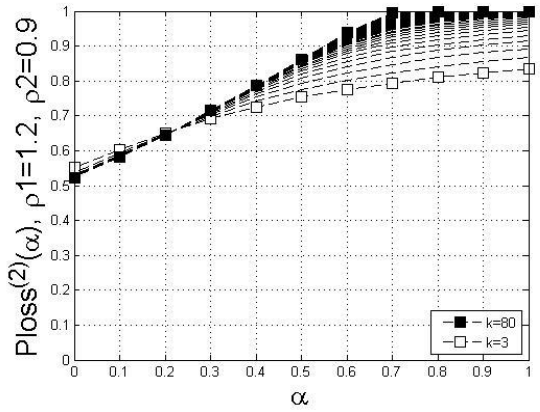


Figure 4. Loss probability of non-priority packets with buffer size K=3-80

In [2] and [3], we have calculated different variable like average queue length of priority packets and the relative time that the priority packet spend in queueing system. So the α parameter is strong enough to influence the filling of the queue.

In [3] it is shown that proposed queueing mechanism provide a wide range of control feature by randomized push-out parameter α and buffer size k. According to the packet's mark (Forbidden, Priority, Background) the period that packet spend in queue can vary from 1 to 10^{14} times, which can be used to enforce requested QoS for traffic. For highly congested network the priority type is much less important, than the push-out mechanism and the value of α parameter. The push-out mechanism allows enforcing access policy using traffic priority mechanism.

By choosing α parameter we can change the time that packets spend in the firewall buffer, which allows to limit access possibilities of background traffic. So by decreasing the priority of background VCs and increasing the push-out probability α we can reduce the VC throughput to low level without interrupting it.

The most wide range of control can be reached in intermediate environment conditions when linear law of the losses has already been broken, but the saturation zone has not been reached yet. Numerical experiment [3] has been made to detect conditions in which ρ_1 varied over a wide range from 0,1 to 2,5, and few fixed values for ρ_2 .

III. PRACTICAL APPLICATION AND FUTURE DEVELOPMENT

Good example of practical application of such mechanism is the problem of controlling remote robotic object, which telemetry data and a video stream are transmitted on global networks [5]. In this case, control commands are transmitted by TCP, and a video stream data are transmitted by UDP. A mean values of throughput of our robotic object: throughput of TCP channel (control and telemetry packets) ~100Kb/s, throughput of UDP video stream ~1,2Mb/s.

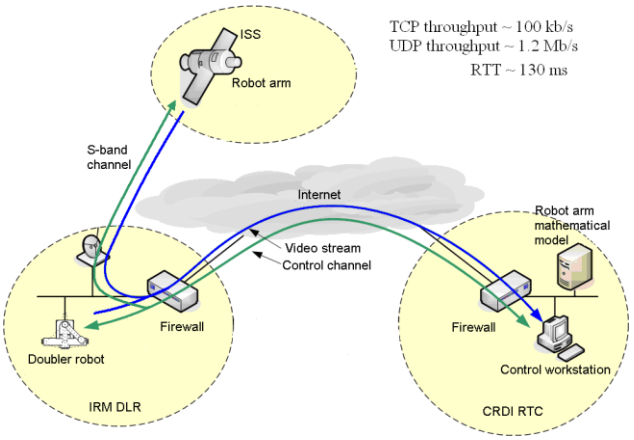


Figure 5. The scheme of space experiment "Kontur"

In a considered example from Figure 5 (ROKVISS (Robotic Component Verification on the ISS) mission and space experiment "Kontur" [6][7]), the choice of a priority of service and loss-probability of a priority packet α allows to balance such indicators of functioning of a network, as loss-probability of control packets $p_{loss}^{(1)}$ and quality of video stream for various conditions of a network environment. The parameter α can vary for delay minimization in a control system's feedback.

The given problem is important for interactive control of remote real-time dynamic objects, in a case when the complex computer network is the component of a feedback control contour, therefore minimization of losses and feedback delays, is the important parameter characterizing an effectiveness of control system.

The same traffic management system is used in space experiment "Kontur-2", where the operator is onboard ISS and robotic system is on Earth surface in Saint-Petersburg, Russia and in Munich, Germany (see Figure 6).

During the space experiments "ROKVISS" and "Kontur" that were minutely considered [1][6]-[8], it was established that an important component of effective remote control for robotic object is a high degree of "dipping" of the operator in the robot's functioning environment [8]. Thus a telepresence for operator in such an environment is ensured by outputting a video stream coming from the camera of the robot into a workplace of operator-cosmonaut. Simultaneously the tactile capabilities of the robot are reproduced by the special joystick, which is connected with the operator through man-machine interfaces with the force-torque feedback. That is why the development of the space experiment "Kontur" was considered a study on the effect of weightlessness on the opportunity of operator-cosmonaut to control remote robots using force-torque joystick. The necessity to send the equipment, which will interact with the cosmonauts onboard ISS gives us new requirements for reliability and security for all systems' components, including communication channels.

An actual scientific and technical problem of the new space experiment "Kontur-2" is the creation of methods and development of the technology for remote control onsurface robots and robotics groups from orbiting spacecraft for solving planetary exploration.

The ability to effectively control robots on the surface of planets from a manned orbital spacecraft is determined by the following factors:

- reliability of telecommunication channels used for transmitting control commands, telemetry and video data between an operator and robot, their capacity for reconfiguration and scalability;
- performance of communication channels in order to minimize the delay of transmitted data;
- adequate response of the operator in weightlessness to the impact from the joystick with force-torque feedback, taking into account the time delays and discontinuity of video feedback received from the controlled robotic object.

The study of these factors significantly differs this work from the previous ones [7][9].

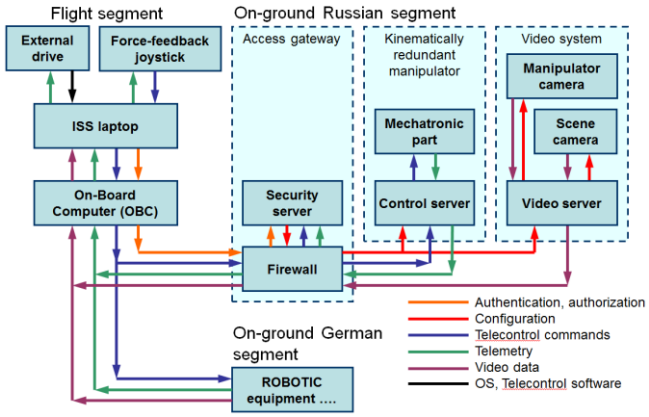


Figure 6. Functional scheme of space experiment "Kontur-2"

In future this method of preemptive access management could be used in new joint space experiment "Kontur-3" (New international space experiment) that will be carried out on ground and on-board the ISS, in order to research efficiency and security of robotic operations in space and ground environments, including the configuration of robotic control systems as a part of robotic communication network [5]. The joint experiments will focus on the analysis of how well astronauts can operate complex robotic systems based on operation networks with mobility and manipulation capability from within the highly constrained ISS and

 ISBN: 978-1-61208-427-5

micro-gravity environment. Multiple human-robot interfaces will be used in combination, while simulating realistic robotic remote operations with round-trip time communication conditions representative of future human planet exploration missions. The structure of data transmission of future experiment is presented in Figure 7.

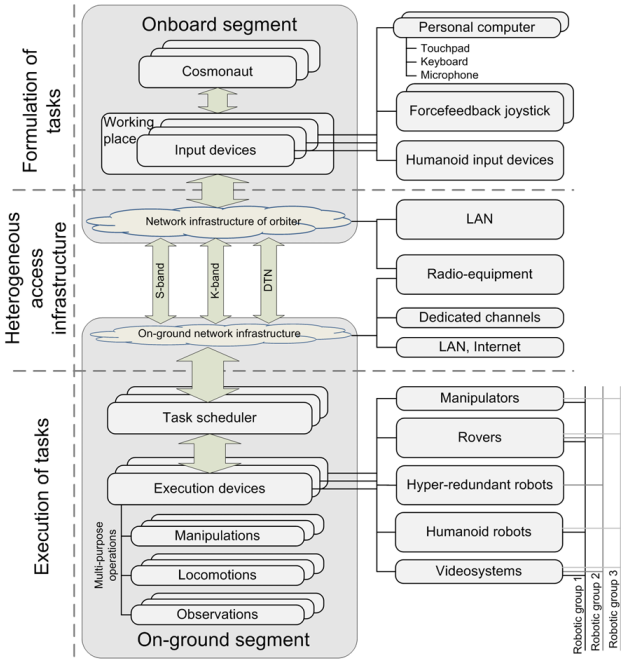


Figure 7. Data streams in future space experiment "Kontur-3"

For communication experiments, the primary focus will be on the usage of real-time duplex commanding, in combination with Delay Tolerant Network (DTN) approaches. Real-time channel will have low delay (15-20 ms) and high throughput (4 Mb/s), but the connection would be established only when the space station is in the radio-optical range (7-10 min). DTN channels have high delay and low throughput, but function for 24/7.

To enforce access policy and provide information security we had to consider the use of various communication channels in future experiments. That is why the access gateway has two levels of filtering: static and dynamic. Static level allows us to control unchanging policy requirements (such as the use of white lists IP addresses and user authentication process). Dynamic control checks suspicious actions of the permitted users, as well as controls transmitted data, taking into account the used transmission channels and the QoS requirements. Some components of the system have been worked out by us in the framework of a contract with Ford Motor Company [8][10]-[12].

The basis of every robotic operation network is high-performance cloud, which is used to decompose the complex task from operator and to monitor its implementation by each robot. So robotics objects within multipurpose operation network would execute the programs and interact without human involvement. However there would be always situations when the robot could not make a decision by its own. In that case the human-operator will have two main opportunities:

1) remote telecontrol through real-time channel;
2) to send new program through DTN.

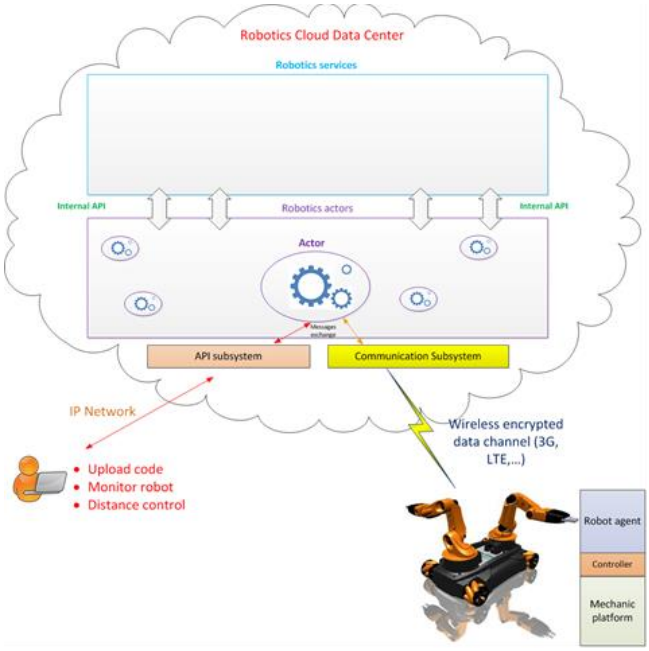


Figure 8. Robotic cloud platform with heterogeneous computing resources

Heterogeneous cloud platform provides not only remote access for computing resources or applications, but also intelligent services. Using the resources of modern cloud-based engineering centers it is possible to create equivalent social networks that bring together multiple agents to perform coordinated actions, computation, verification of test results based on the use of different materials, virtual prototyping, and data visualization. These problems, from the point of view of the computational algorithms, can be combined into chains, which form a network of operations. Their implementation is provided within a heterogeneous cloud. The components of the platform (see Figure 8), based on the OpenStack, include: IaaS cloud class segment, computing infrastructure within the cluster, the specialized high-performance hybrid system based on reconfigurable computing nodes.

Virtualization has changed the approach of deploying, managing, and using enterprise resources by providing new opportunities for consolidation and scalability of computational resources available to applications; however, this leads to the emergence of new threats posed by the complexity and dynamic nature of the process of resources provisioning. These threats can lead to the formation of cascade security violations, which traditional data protection systems are unable to deal with. The existing approaches such as “Scan and Patch” do not work in a cloud environment — network scanners cannot track changes of

 ISBN: 978-1-61208-427-5

resources configuration in real time. These approaches do not accurately identify the change in level of risk and take steps to block dynamically emerging threats.

To solve the problem of controlling access in the cloud, it is necessary to continuously monitor resources, and it cannot be achieved without the automatic generation of rules for filtering and firewall log files analysis. Information security management products in a dynamic cloud environment should include mechanisms that provide: total control over processes for deploying virtual machines; proactive scanning the virtual machines for the presence of vulnerabilities and configuration errors; tracking the migration of virtual machines and system configuration to control access to resources. Therefore, within the space experiment "Kontur-2" series of measures are set out to improve information security resources, namely:

- Enhanced Control of virtual machines. Virtual machines as active components of the service are activated in the cloud application random moments, and Administrator cannot activate or deactivate a virtual machine until the security scanner checks the configuration and evaluates the security risks.
- Automatic detection and scanning. Information security services are based on discovery of vulnerabilities in the computing environment. This discovery in turn is based on the current virtual machine configurations and on reports of potential threats that come from trusted sources, such as antivirus update servers.
- Migration of virtual machines. Proactive application migration is an effective method to control security. Each of these data will have its own priority level. The number of priorities could be increased by using the recursive application of prioritization method described above.

IV. CONCLUSION

The paper illustrates one of the possible applications of access control and traffic management approach in the tasks of robotic remote control in space experiment “Kontur-2”.

Proposed model considers computer network as the set of VCs, which throughput is easy controlled by proposed classification procedure and algorithm that divides the set of non forbidden VCs in two subsets: non forbidden priority connections and non-forbidden non-priority ones or background connections.

In this paper, we considered in detail the preemptive queueing mechanism, which provides a wide range packet loss probability ratio using flexible randomized push-out algorithm. The most interesting result obtained in congested network allows keeping priority VC throughput near the requested value, which is important for specific space experiment onboard ISS.

Described a practical application example of proposed model in joint space experiment “Kontur-2” onboard ISS where several types of operations are serviced by access gateway in robotic communication network.

Proposed an architecture of access gateway for robotic cloud platform with heterogeneous computing resources that expected to be used in future space experiments onboard ISS.

ACKNOWLEDGMENT

The reported study was partially supported by RFBR, research project No. 15-29-07131 ofi_m.

REFERENCES

[1] V. Zaborovsky, O. Zayats, and V. Muljukha, "Priority Queuing with Finite Buffer Size and Randomized Push-out Mechanism" // Proceedings of the Ninth International Conference on Networks ICN 2010, pp.316-321.

[2] A. Ilyashenko, O. Zayats, V. Muliukha, and L. Laboshin, "Further Investigations of the Priority Queuing System with Preemptive Priority and Randomized Push-Out Mechanism" // Internet of Things, Smart Spaces, and Next Generation Networks and Systems, Lecture Notes in Computer Science, vol. 8638, pp.433-443.

[3] V. Muliukha, A. Ilyashenko, O. Zayats, and V. Zaborovsky, "Preemptive queueing system with randomized push-out mechanism", Communications in Nonlinear Science and Numerical Simulation, Volume 21, Issues 1–3, April 2015, pp.147-158.

[4] V. Zaborovsky, A. Gorodetsky, and V. Muljukha, "Internet Performance: TCP in Stochastic Network Environment" // Evolving Internet, 2009. INTERNET '09. First International Conference on, pp.21–26.

[5] V. Zaborovsky, O. Zayats, V. Muljukha, and A. Ilyashenko, "Cyber-Physical Approach to the Network-Centric Robot Control Problems" // Internet of Things, Smart Spaces, and Next Generation Networks and Systems, Lecture Notes in Computer Science, vol. 8638, pp.619–629.

[6] http://www.nasa.gov/mission_pages/station/research/experiments/15.html [retrieved: July, 2015].

[7] V. Zaborovsky and V. Mulukha, “Access Control in a Form of Active Queuing Management in Congested Network Environment” // Proceedings of The Tenth International Conference on Networks (ICN 2011), St. Maarten, The Netherlands Antilles, January 23-28, 2011 – Published by XPS. – 2011. – pp.12-17.

[8] V. Muliukha, V. Zaborovsky, and S. Popov, “Security of Vehicular Networks: Static and Dynamic Control of Cyber-Physical Objects” // SECURWARE 2014 - 8th International Conference on Emerging Security Information, Systems and Technologies, pp.56-61.

[9] A. Albu-Schaffer, W. Bertleff, B. Rebele, B. Schafer, K. Landzettel, and G. Hirzinger, “ROKVISS - robotics component verification on ISS current experimental results on parameter identification” // IEEE International Conference Robotics and Automation. – 2006. – pp. 3879-3885

[10] L. Kurochkin, S. Popov, M. Kurochkin, V. Glazunov, “Instrumental environment of multi-protocol cloud-oriented vehicular mesh network” // ICINCO 2013 - Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics, pp. 568-574.

[11] V. S. Zaborovskiy, A. A. Lukashin, A. V. Vostrov, S. G. Popov “Adage mobile services for ITS infrastructure” // 13th International Conference on ITS Telecommunications, ITST 2013, pp. 127-132.

[12] L. Kurochkin, S. Popov, M. Kurochkin, V. Glazunov, “Hardware and software equipment for modeling of telematics components in intelligent transportation systems” // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2014, pp. 598-608.

 ISBN: 978-1-61208-427-5

Secure Scrum: Development of Secure Software with Scrum

Christoph Pohl
and Hans-Joachim Hof

MuSe - Munich IT Security Research Group
Munich University of Applied Sciences
Email: christoph.pohl0@hm.edu, hof@hm.edu

***Abstract*—Nowadays, the use of agile software development methods like Scrum is common in industry and academia. Considering the current attacking landscape, it is clear that developing secure software should be a main concern in all software development projects. In traditional software projects, security issues require detailed planning in an initial planning phase, typically resulting in a detailed security analysis (e.g., threat and risk analysis), a security architecture, and instructions for security implementation (e.g., specification of key sizes and cryptographic algorithms to use). Agile software development methods like Scrum are known for reducing the initial planning phases (e.g., sprint 0 in Scrum) and for focusing more on producing running code. Scrum is also known for allowing fast adaption of the emerging software to changes of customer wishes. For security, this means that it is likely that there are no detailed security architecture or security implementation instructions from the start of the project. It also means that a lot of design decisions will be made during the runtime of the project. Hence, to address security in Scrum, it is necessary to consider security issues throughout the whole software development process. Secure Scrum is a variation of the Scrum framework with special focus on the development of secure software throughout the whole software development process. It puts emphasis on implementation of security related issues without the need of changing the underlying Scrum process or influencing team dynamics. Secure Scrum allows even non-security experts to spot security issues, to implement security features, and to verify implementations. A field test of Secure Scrum shows that the security level of software developed using Secure Scrum is higher then the security level of software developed using standard Scrum.**

Keywords–Scrum; Secure Scrum; Security; Secure Software Development; SDL

I. INTRODUCTION

Nowadays, software is all around us, even refrigerators now have network support and run a whole bunch of software. As software is so ubiquitous today, software bugs that lead to successful attacks on software systems are becoming a major hassle. Hence, modern software development should focus on SECURE software. At the moment, Scrum [1][2] is a very popular software development framework. This paper presents Secure Scrum, an extension of the Scrum framework that helps developers, even non-security experts, to develop secure software.

Scrum groups developer in small developer team that have a certain autonomy to develop software. It is assumed that all developers can implement all tasks at hand. Software is incrementally developed in so called sprints. A sprint is a fixed period of time (between 2 and 4 weeks). During a sprint, the team develops an increment of the current software version, typically including a defined number of new features or functionality, which are described as user stories. User stories are used in Scrum to document requirements for a software project. All user stories are stored in the Product Backlog. During the planning of a sprint, user stories from the Product Backlog are divided into tasks. These tasks are stored in the Sprint Backlog. A so called Product Owner is the single point of communication between customer and developer team. The Product Owner also prioritizes the features to implement. Standard Scrum does not include any security-specific parts.

One major driver of software security in Secure Scrum is the identification of security relevant parts of a software project. The security relevance is then made visible to all team members at all times. This approach is considered to increase the security level because developers place their focus on things that they had evaluated themselves, which they fully understand, and when their prioritization of requirements does not differ from prioritization of others [3][4].

Secure Scrum aims on achieving an appropriate security level for a given software project. The term ”appropriate” was chosen to avoid costly over engineering of IT security in software projects. The definition of an appropriate security level is the crucial point in resource efficient software development (e.g., time and money are important resources during software development). For the definition of an appropriate security level, Secure Scrum relies on the definition in [5]: Software needs to be secured until it is no longer profitable for an intruder to find and exploit a vulnerability. This means that an appropriate security level is reached once the cost to exploit a vulnerability is higher then the expected gain of the exploit. So, Secure Scrum offers a way to not only identify security relevant parts of the project but to also judge on the attractiveness of attack vectors in the sense of ease of exploitation.

Related to the identification of security issues, the developers need to implement features to avoid these potential security risks. In Scrum, each team member is responsible for the completeness of his solution (Definition of Done). However, there is a huge number of choices of methodologies to verify completeness. This means that a team member can use any method for verification (same as with normal tests, Scrum

 ISBN: 978-1-61208-427-5

does not tell the developer how to test). Secure Scrum helps developers to identify appropriate security testing means for security relevant parts of a software project.

One last challenge solved by Secure Scrum is the availability of know how when needed. Secure Scrum assumes that the vast majority of requirements should and could be handled by the team itself to keep many benefits of Scrum. However, for some security related issues, it could be necessary or more cost effective to include external resources like security consultants in the project. Secure Scrum offers a way to include these external resources into the project without breaking the characteristics of Scrum and with little overhead in administration.

The rest of this paper is structured as follows: The following section summarizes related work. Section III shows the design of Secure Scrum in detail. Secure Scrum is evaluated in a field test in Section IV. Section V summarizes the findings of this paper.

II. Related Work

There are several methods for achieving software security, e.g., Clean Room [6], Construction by Correctness [7], CMMI-DEV [8][9], etc. However, these methods cannot be used in Scrum as they clash with the characteristics of agile software development and specifically Scrum. Construction by Correctness [7] for example, advocate formal development in planning, verification and testing. This is completely different to agility and flexible approaches like agile methodologies. Other models like CMMI-DEV [8][9] can deal with agile methods, but they are process models. The main difference is that CMMI focuses on processes and agile development on the developers [9]. This means that Scrum and other agile methodologies are developer centric, while CMMI is more process oriented. Concepts like Microsoft SDL [10] are designed to integrate agile methodologies, but is also self-contained. It can not be plugged into Scrum or any other agile methodology. Scrum focuses on rich communication, self-organisation, and collaboration between the involved project members. This conflicts with formalistic and rigid concepts.

To sum it up, the major challenge of addressing software security in Scrum is not to conflict with the agility aspect of Scrum.

S-Scrum [11] is a "security enhanced version of Scrum". It modifies the Scrum process by inserting so called spikes. A spike contains analysis, design and verification related to security concerns. Further, requirements engineering (RE) in story gathering takes effect on this process. For this, the authors describe to use tools like Misuse Stories [12]. This approach is very formalistic and needs lot of changes to standard Scrum, hence hinders deployment in environments already using Scrum. Secure Scrum in contrast does not change standard Scrum.

Another approach is described in [13]. It introduces a Security Backlog beside the Product Backlog and Sprint Backlog. Together with this artifact, they introduce a new role. The security master should be responsible for this new Backlog. This approach introduces an expert, describes the security aware parts in the backlog, and is adapted to the Scrum process. However, it lacks flexibility (as described in the introduction) and does not fit naturally in a grown Scrum team. Also, the introduction of a new role changes the management of the project. With this approach it is not possible to interconnect standard Scrum user stories with the introduced security related stories. Secure Scrum in contrast keeps the connect between security issues and user stories of the Product Backlog respectively tasks of the Sprint Backlog.

In [14] an informal game (Protection Poker) is used to estimate security risks to explain security requirements to the developer team. The related case study shows that this is a possible way to integrate security awareness into Scrum. It solves the problem of requirements engineering with focus on IT Security. However, it does not provide a solution for the implementation and verification phase of software development, hence it is incomplete. Secure Scrum in contrast provides a solution for all phases of software development.

Another approach is discussed in [15]. An XP Team is accompanied by a security engineer. This should help to identify critical parts in the development process. Results are documented using abuse stories. This is similar to the definition in [16]. This approach is suitable for XP-Teams but not for Scrum.

To sum it up, none of the related work mentioned above integrates well into Scrum, allows for easy adaption for teams already using standard Scrum, and focuses on all phases of software development. Secure Scrum in contrast solves all of these problems. The design of Secure Scrum is described in detail in the following.

III. Design of Secure Scrum

Secure Scrum consists of four components:

- Identification component
- Implementation component
- Verification component
- Definition of Done component

These four components are put on top of the standard Scrum framework. Secure Scrum influences six stages of the standard Scrum process as can be seen in Figure 1.

The identification component is used to identify security issues during software development. Security issues are marked in the Product Backlog of Scrum. The identification component is used during the initial creation of the Product Backlog as well as during Product Backlog Refinement, Sprint Planning, and Sprint Review.

The implementation component raises the awareness of the Scrum team for security issues during a sprint. The implementation component is used in Sprint Planning, as well as during the Daily Scrum meetings.

The verification component ensures that team members are able to test the software with the focus on IT Security. The verification component gets managed within the Daily Scrum meeting.

The Definition of Done component enables the developers to define the Definition of Done for security related issues as postulated in standard Scrum.

These four components of Secure Scrum are described in detail in the following subsections.

 ISBN: 978-1-61208-427-5

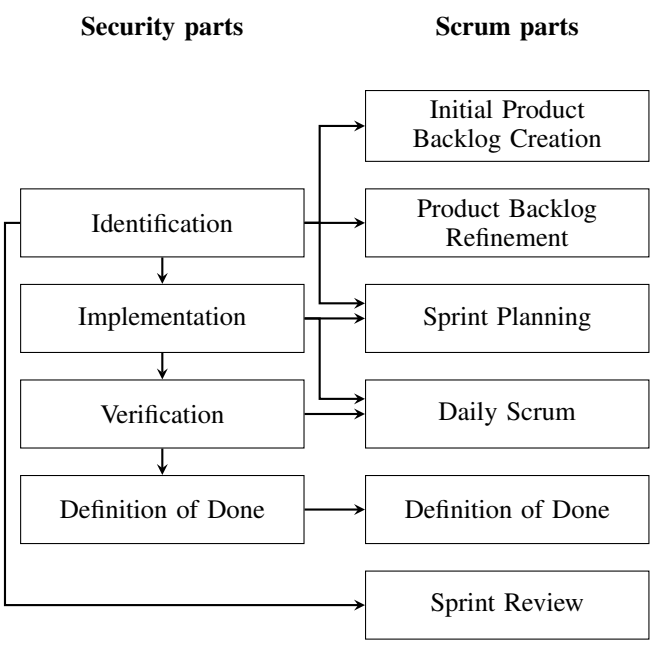


Figure 1. Integration of Secure Scrum components into standard Scrum

A. Identification Component

The identification component is used to identify and mark security-relevant user stories. Secure Scrum takes a value-oriented approach to security as described in the Introduction. It focuses security implementation effort on parts of the emerging software that are of high value for the stakeholders. The identification component of Secure Scrum is used during initial Product Backlog creation, during Sprint Planning, as well as during Product Backlog Refinement.

In a first step, stakeholders (may be represented by the Product Owner) and team members rank the different user stories according to their loss value. The loss value of a user story is not the cost of development neither the benefit of the functionality that implements the user story. The loss value of a user story is the loss that may occur whenever the functionality that implements the user story gets attacked or data processed by this functionality gets stolen or manipulated. For example one can formulate "Whenever someone will get access to these data, our company will have high damage". Even better the cost gets listed with a numerable value like USD or Euro.

In a next step, stakeholders and team members evaluate misuse cases and rank them by their risk.

At this point, it can be useful to incorporate external security expertise to moderate by asking the right questions and proposing security aware user stories.

After finalization of the identification component, team members and stakeholders have a common understanding of security risks in the Product Backlog. To document this understanding in the Product Backlog, Secure Scrum uses so called *S-Tag*s. Figure 2 shows the basic principle of an *S-Tag*. An *S-Tag* consists of one or more *S-Mark*s, a Backlog artifact, and a connection between the Backlog artifact and one or more *S-Tag*s. An *S-Tag* identifies Product Backlog items that have security relevance with a Marker called *S-Mark*. This ensures

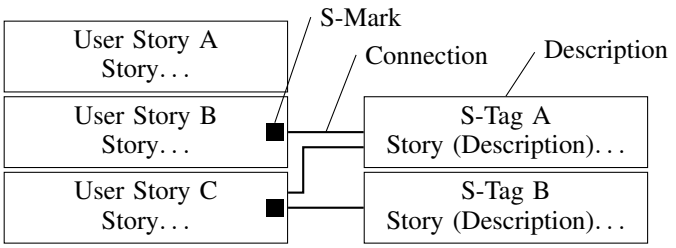


Figure 2. Usage of S-Tags to mark user stories in the Product Backlog and to connect user stories to descriptions of security related issues.

that security sensitive items in the Product Backlog are visible at all times. The technology behind the *S-Mark* is negligible (it can be a red background, a dot, or something else), it only must ensure that a Product Backlog item with security relevance contrasts to other Backlog items.

An *S-Tag* describes a security concern. A detailed description of the security issue helps the Scrum team to understand the security concern. The description of the security concern itself can be formulated in a separate Backlog item. This can be a user story, misuse story, abuse story, or whatever a team decides to use as description technology. The description may include elements from a knowledge base that gives advice on how to deal with this specific security concern. If such a knowledge base is maintained over the course of several projects, it is very likely a valuable source of information for the Scrum team.

An *S-Tag* links one security concern to one or more Backlog items. A security concern is any security related problem, attack vector, task, or security principle that should be considered during implementation. One-to-many-connections between security concern and affected Product Backlog items allow for grouping of items that share the same security concern (and hopefully may use the same security mechanisms) as well as expressing security on a high level. To express the connections, unique IDs can be used.

B. Implementation Component

The Scrum framework has a focus on implementation. Thus, during implementation every team member needs to be aware of the top priority topics of the project. This means that most of the requirements (functionalities) are described in the Product Backlog. This includes the *S-Tag*s. To ensure that security concerns are visible in daily work, they must be present in the Sprint Backlog.

Usually, a sprint implements a subset of functionalities (for example user stories). During a sprint, some user stories are broke down to tasks (or similar conceptual parts). Whenever a user story is marked with an *S-Mark*, the corresponding *S-Tag* must also be present in the correspondding sprint. An *S-Tag* can be handled like any other Backlog item. But whenever an *S-Tag* gets splitted into tasks, the tasks must also be marked with an *S-Mark* and connected to the original *S-Tag*. This ensures that developers are always aware of the original security concern and it can be linked back to the origin description.

This approach determines that the interconnection will enhance the awareness of the developer for the security problems.

 ISBN: 978-1-61208-427-5

C. Verification Component and Definition of Done Component

Not only *S-Tag*s help developers to be aware of security relevance of user stories, they can also be used to identify requirement for the verification of the emerging software. In the first place, *S-Tag*s clearly identify parts of the emerging software that need security verification. In the second place, *S-Tag*s are useful to estimate the effort for verification.

Secure Scrum proposes two different approaches for verification and therefore the Definition of Done. For further simplification, the term "task" is used for some work that is performed by one developer in one sprint and that needs one Definition of Done. Whenever the verification process (whatever the developer or team chooses to use) for one task can be performed during the same sprint and from the same developer, the verification must be part of the task. This ensures that the verification must be part of the Definition of Done. However, it is possible that a developer does not have the required knowledge for verification, or the verification needs external resources, extra time for testing, or anything else that hinders an immediate verification. In this case, the verification cannot be part of the Definition of Done. In such cases, a new task must be created which inherits only the verification part. This new task must be marked with an *S-Mark* and should be connected to the original *S-Tag*, together with the original task. Then, the developer can define the Definition of Done without the verification, hence a Definition of Done compatible to standard Scrum is available.

The proposed approach for the definition of the Definition of Done ensures that the connection between an *S-Mark* and its corresponding *S-Tag* keeps existing throughout the project, hence no security concern can get lost or stay untested.

D. Integration of External Ressources

IT security knowledge may be rare in a Scrum team or special knowledge not present in the Scrum team may be necessary for certain parts of the emerging software (e.g., implementation and testing of cryptographic algorithms). Secure Scrum offers ways to include external resources (e.g., security consultants) in all components of Secure Scrum. External resources could have one or more of the following three functions:

- Enhance knowledge
- Solve challenges
- Provide external view

These three functions are described in the following.

Enhancing knowledge: This function includes security-related training for the Scrum team to help them to gain a better understanding of a specific security-related area. Doing so on the job during a project offers a chance to teach IT security with a specific example at hand (e.g., a certain *S-Tag* that is linked to many user stories) and may be more efficient than security training during two projects. Training may be necessary for aspects that are not part of everyday work, e.g., the usability of security mechanisms [17], [18].

Solving Challenges: Some *S-Tag*s represent hard security challenges that require special expertise or special experience, such that it is more cost efficient to let external resources solve this challenge. To avoid breaches in Scrum, it is necessary that these external solutions can be handled like a tool, a well defined part of development, a framework, or a "black box", which is ready to use. This means that this external solution should be encapsulated and therefore does not influence Scrum or the Scrum team. For example, this can be a functional part of software (with special IT Security concerns) or parts of the project which can be used with an API by the Scrum team. Another challenge is the integration of external services like penetration testing into the development process. One way to do so is that external resources provide test cases (e.g., for Metasploit [19]) that can be used for every branch of the emerging software at any time. Results of tests can be documented as artifacts in the Backlog. Then they can be handled like any other change request.

Providing external view: One major part in IT Security is to recognize ways to exploit the own system. In other words, one must think like an attacker to recognize potential attack vectors. Usually, it is easier for an outsider to spot potential weaknesses of a system than it is for the developer of a system. Hence, external resources may introduce a valuable external viewpoint on a project. When using the identification component of Secure Scrum, an external consultant can be helpful to point the team to security concerns. When using the implementation component, external resources can be helpful in the sprint planning. When using the verification component, an external consultant can help to create tests for security concerns. These interventions by external resources should not be part of the normal Scrum processes, the external resource should only help to ask questions (in the meaning of: he should show relevant concerns in scope of IT Security). In conclusion, the external resource should help to set focus on problems the team is not aware of.

IV. Evaluation

The evaluation presented in this paper focuses on the following questions: is Secure scrum a practicable approach to develop secure software? Is Secure Scrum easy to understand? Does Secure Scrum increase the security level of the developed software?

As test setting, 16 developers were asked to develop a small piece of software. The developers were third year students in computer sciences and business informatics (BSc). They were not aware that they are part of this evaluation. The students showed programming skills that were on the usual level of a third year bachelor student. No participant attended a specialized course in IT Security before beside the compulsory lecture in IT Security (basic level) in the second year of the bachelor. All developers had average theoretical knowledge about Scrum. Only two students had practical experiences (less than 2 months) with Scrum. No one had practical experiences in IT Security.

The developers were divided into three groups.

1) Team 1 (T1): The Anarchist group: They could manage themselves as they like, except using Scrum
2) Team 2 (T2): The Scrum group: They should use standard Scrum
3) Team 3 (T3):The Secure Scrum group: They should use Secure Scrum.

To avoid influences on the evaluation, teams 1 and 2 thought that team 3 also uses standard Scrum. All groups got a list of six requirements for a new software product.

 ISBN: 978-1-61208-427-5

TABLE I. Results of the evaluation of the efficiency and effectiveness of Secure Scrum

#	Metric	T1	T2	T3
1	Lines of Code	1149	758	458
2	Number of Basic Requirements	6	6	6
3	Number of additional Requirements defined	0	1	8
4	Number of Basic Requirements documented	0	6	6
5	Number of Basic Requirements implemented	6	5	4
6	Number of Requirements documented	0	7	14
7	Number of Requirements implemented	6	6	9
8	Number of Vulnerabilities sp	18	12	3
9	Group size	6	5	5

They were asked to develop a prototype for a social network with the following features: registration, login, logout, personal messages, wall messages, bans, friend lists, and further more. Each group had only one week to develop this prototype using Java and a preconfigured spring framework template (based on BREW [20]). Each group was asked to develop a piece of software including as many requirements as possible (they knew that it was impossible to implement all requirements for the final version of the software considering the harsh time constraints). They were also told that they need to "sell" their prototype on the last day of the experiment in front of a jury. In fact they should learn how to present their prototype and act like a team that wants to have a contract for further development. This should ensure that every team needs to define for itself the selling points of their prototype.

Team 3 has a short briefing of about one hour about Secure Scrum. Every team is advised to make a proper documentation. This includes all produced artifacts, the sources, and a short description of their development process.

Table I summarizes some basic findings of the experiment.

All three teams had a rough definition of the six basic requirements which should be implemented. They were told that whenever the requirements list should be enhanced to deal with the 6 requirements given by the customer, they are free to define new requirements. Team 1 did not define any new requirements. Team 2 defined one new requirement to enhance performance. Team 3 defined 8 new requirements that had a focus on IT Security. These requirements are an excerpt of the descriptions for the *S-Tag*. Overall, they defined 29 new stories focused on IT Security. This shows that even with beginner skills in computer sciences and low skills in IT Security, it is possible to define a high amount (compared to the original requirements) of security related requirements. It also shows that it is possible to describe the most problematic vulnerabilities or problems with the help of risk identification.

Metrics $4-7$ of table I are used to evaluate if the teams documented all requirements and how many of the requirements were implemented. This shows that the teams did not take care of any further requirements when not specified by the customer. This sounds trivial, but it also shows that the developer did not take care of IT Security when not specified. The Secure Scrum team (team 3) is the only team that did not implement all basic requirements. Instead, they obviously prioritized some of the security requirements over the basic requirements as some of the additional requirements were implemented. This finding shows that Secure Scrum helps to put focus on software security.

TABLE II. Results of the evaluation of the practicability of Secure Scrum

#	Metric	Team 2	Team 3
1	Number of requirements	7	14
2	Number of user stories	7 (13)	14 (62)
3	Number of tasks	18	35
4	Number of user stories with *S-Mark*	-	14
5	Number of tasks with *S-Mark*	-	8(35)

Metric 8 shows the number of security problems that were created by the developers. The number of security problem is calculated as follows: Let sl be a vulnerability listed in the OWASP Top 10 list OTT ($sl \in OTT$) . The OWASP Top 10 project lists the most common security vulnerabilities for web applications, e.g., Injection, Broken Authentication and Session Management, Cross-Site Scripting (XSS), Insecure Direct Object References, Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross-Site Request Forgery, Using Components with Known Vulnerabilities, and Unvalidated Redirects and Forwards. Let OS be the complete source code of the developed software and SC the part of the software written by the students ($SC \subset OS$). Let cf be a Java function. Let $cpf(sl)$ be a function that counts the amount of sl for one cf. By definition, $cpf(sl)$ increments a vulnerability counter by one whenever the current function is the source ms function for a vulnerability. A function cf is considered as a source ms whenever $cf \in SC$ and when the function is the reason for the vulnerability or it calls a function cf_1 where $cf_1 \notin SC$ and cf_1 is the reason for the flaw. The amount of vulnerabilities sp is the sum of all $cpf(cf)$. Such a definition of the number of security problems only counts code that is responsible for vulnerabilities of a software system. It also takes into consideration the use of flawed code. For example, when a developer creates an SQL statement with a potential SQL Injection flaw, the function holding the database call with this statement is regarded as the reason of the vulnerability. The results of the evaluation shows, that team 1 and team 2 had a high amount of vulnerabilities in their software (team 1: 18, team 2: 12). Both teams built software exploitable by SQL Injection, XSS, CSRF, and had a vulnerable session management. Team 3 had significantly less vulnerabilities. This shows, that the use of Secure Scrum increase the security level of the developed software.

The first metric (Lines of Code (LOC)) shows the amount of code which was generated during the week. There are significant differences between the three teams. The teams that identified additional requirements (performance (team 2) and security (team 3)) were not as productive as the other teams. This shows the overhead that comes with a broadened focus on software quality, especially on non-functional requirements.

To evaluate ease of use and practicability of Secure Scrum, the documentation of the Scrum teams was evaluated. The documentation consists of the Backlogs and a timetable. Table II summarizes the results of this evaluation.

The numbers in braces are the total amount of user stories. The aggregated number (not in braces) shows the amount of user stories when grouped together. This means a group of user story is a "bigger" user story which reflects a requirement. Team 2 broke down every user story to a different task.

 ISBN: 978-1-61208-427-5

Team 3 broke down tasks for only the stories that they also implemented. This is why they defined more user stories than tasks. Team 3 found for every user story some security concerns, this is why they tagged all user stories. Metric 5 shows that all tasks also had *S-Mark*s, overall they had 8 different groups in the tasks. Team 3 decided to create the links by grouping, they simply used red cards for the descriptions to show security problems (*S-Mark*). This also shows that the proposed tools are simple enough to adapt them very fast in a Scrum process.

In conclusion, the evaluation shows that Secure Scrum is able to improve the security level of the developed software. Secure Scrum is easy to understand, can be used in practice, and is even suitable for teams that have no deepened security knowledge. The evaluation also shows that it is possible to have a proper documentation through all stages of the experiment. The tools of Secure Scrum harmoniously blend into the standard Scrum toolset without the need of much overhead for training.

V. Conclusion

This paper presents Secure Scrum, an extension of the software development framework Scrum. Secure Scrum enriches Scrum with features focusing on building secure software. One of the main contributions of Secure Scrum are *S-Tag*s, a way to annotate Backlog items with security related information. Secure Scrum was evaluated in a small software development project. The evaluation shows that Secure Scrum can be used in practice, is easy to use and understand, and improves the level of software security.

References

[1] K. Beck, M. Beedle, K. Schwaber, and M. Fowler, "Manifesto for agile software development," retrieved: 07, 2015. [Online]. Available: http://www.agilemanifesto.org/

[2] K. Schwaber, "SCRUM development process," in Business Object Design and Implementation, D. J. Sutherland, C. Casanave, J. Miller, D. P. Patel, and G. Hollowell, Eds. Springer London, pp. 117–134.

[3] C. Riemenschneider, B. Hardgrave, and F. Davis, "Explaining software developer acceptance of methodologies: a comparison of five theoretical models," IEEE Transactions on Software Engineering, vol. 28, no. 12, Dec. 2002, pp. 1135–1145.

[4] L. Vijayasarathy and D. Turk, "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," Information and Software Technology, vol. 54, no. 2, Feb. 2012, pp. 137–148.

[5] C. Herley, "Security, cybercrime, and scale," Communications of the ACM, vol. 57, no. 9, Sep. 2014, pp. 64–71.

[6] H. D. Mills and R. C. Linger, "Cleanroom Software Engineering: Developing Software Under Statistical Quality Control - Encyclopedia of Software Engineering - Mills - Wiley Online Library," 1991.

[7] A. Hall and R. Chapman, "Correctness by construction: developing a commercial secure system," IEEE Software, vol. 19, no. 1, 2002, pp. 18–25.

[8] M. B. Chrissis, M. Konrad, and S. Shrum, CMMI for Development, ser. Guidelines for Process Integration and Product Improvement. Pearson Education, Mar. 2011.

[9] H. Glazer, J. Dalton, D. Anderson, M. D. Konrad, and S. Shrum, "CMMI or Agile: Why Not Embrace Both!" 2008, pp. 1–48.

[10] M. Howard and S. Lipner, The security development lifecycle. O'Reilly Media, Incorporated, 2009.

[11] D. Mougouei, N. F. Mohd Sani, and M. Moein Almasi, "S-scrum: a secure methodology for agile development of web services." World of Computer Science & Information Technology Journal, vol. 3, no. 1, 2013, pp. 15–19.

[12] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," Requirements Engineering, vol. 10, no. 1, Jan. 2005, pp. 34–44.

[13] Z. Azham, I. Ghani, and N. Ithnin, "Security backlog in scrum security practices," in Software Engineering (MySEC), 2011 5th Malaysian Conference in. IEEE, 2011, pp. 414–417.

[14] L. Williams, A. Meneely, and G. Shipley, "Protection poker: The new software security," IEEE Security & Privacy, no. 3, 2010, pp. 14–20.

[15] G. Boström, J. Wyrynen, M. Bodn, K. Beznosov, and P. Kruchten, "Extending XP practices to support security requirements engineering," in Proceedings of the 2006 international workshop on Software engineering for secure systems. ACM, 2006, pp. 11–18.

[16] J. Peeters, "Agile security requirements engineering," in Symposium on Requirements Engineering for Information Security, 2005.

[17] H.-J. Hof, "Towards Enhanced Usability of IT Security Mechanisms - How to Design Usable IT Security Mechanisms Using the Example of Email Encryption," International Journal On Advances in Security, vol. 6, no. 1&2, 2013, pp. 78–87.

[18] H. J. Hof, "User-Centric IT Security - How to Design Usable Security Mechanisms," in The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012), 2012, pp. 7–12.

[19] Rapid7, "Metasploit," 2015, retrieved: 07, 2015. [Online]. Available: http://www.metasploit.com/

[20] C. Pohl, K. Schlierkamp, and H.-J. Hof, "BREW: A Breakable Web Application for IT-Security Classroom Use," in Proceedings: European Conference on Software Engineering Education 2014. ECSEE, 2014, pp. 191–205.

 ISBN: 978-1-61208-427-5

Enhanced Authenticated Encryption Scheme

Dr. Eng. / Jamal Abelfatah Morad Azzam
Research Center , SCA
Ismailia, Egypt.
e-mail:jamalazzam@yahoo.com

Abstract− Cryptography is a vital part in information handling. In this paper we introduce a new scheme for encryption and authentication of the encrypted message. A random number is generated in each encryption process. Both of the random number and the secret key is used to generate the subkeys, a different subkey for each data block. To increase the secrecy, double permutation processes are executed on data blocks in the form of mutation and crossover. Mutation process to be performed at an arbitrary bit number, and crossover is performed at another bit number. Encryption of each data block is dependent on the previous encrypted data blocks, the secret key, and the random number. Also, one way hash function is generated to ensure authenticity of the message. The scheme proves its strength against cryptanalysis.

Keywords − encryption ; decryption ; hash function ; secret key.

I. INTRODUCTION

Cryptography refers almost exclusively to encryption, it is the process of converting ordinary information (called plaintext) into unintelligible gibberish (called ciphertext). Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. A cipher in cryptosystems is a pair of algorithms that create the encryption and the reversing decryption [1][2]. The detailed operation of a cipher is controlled by the algorithm and the key. The objectives are the following items [3]-[12]:

- Privacy or confidentiality.
- Data integrity.
- Authentication.
- Non-repudiation.

There are two types of cryptosystems, one-key (or symmetric key), and two-key (asymmetric key) ciphers. In symmetric key ciphers, the encryption of a plaintext and the decryption of the corresponding ciphertext are performed using the same key. Until 1976 when Diffie and Hellman introduced public-key or two-key cryptography all ciphers were one-key systems [14]. Therefore one-key ciphers are also called conventional cryptosystems.

Conventional cryptosystems are widely used throughout the world today, and new systems are published frequently. There are two types of conventional cryptosystems: stream ciphers and block ciphers.

In stream ciphers, a long sequence of bits is generated from a short string of key bits, and it is then added bitwise modulo 2 to the plaintext to produce the ciphertext. In block ciphers the plaintext is divided into blocks of a fixed length, then they are encrypted into blocks of ciphertexts using the same key. Block ciphers can be divided into three groups: Substitution ciphers, Transposition ciphers and, Product ciphers.

Since its introduction in 1977, the Data Encryption Standard (DES) has become the most widely applied private key block cipher [15]. Recently, a hardware design to effectively break DES using exhaustive search was outlined by Wiener [16]. AES by its turn is subjected to different cryptanalysis that presumes its ability to break AES. [17]-[19]. So, there are a need to secure, and flexible block ciphers with immune encryption.

In this paper, a novel randomized scheme is proposed. It uses beside the secret key a random number. Both of them are implemented to generate different subkeys for different blocks of message. The secret key can be of any chosen size, provided key size modulo block size is zero. In each encryption process a new random number is generated, consequently, increases the resistance to cryptanalysis whatever it is based on differential [16], or linear [21]. A hash function is also used to ensure data authenticity. This scheme has significant strict avalanche criterion (SAC), and keeps cryptographic static and dynamic prosperities of the substitution permutation networks (SPNs).

The rest of this paper is organized as follows. Section II describes how to generate the subkeys, and the message authentication code (MAC), and the proposed algorithm. Section III Addresses the encryption process of a random number, and the MAC number, the mutation and crossover processes for message blocks, and encryption / decryption of data blocks. Section IV provides results of strict avalanche effect compared with other algorithms. Section V addresses an evaluation of the algorithm performance compared with other algorithms. Analysis of the algorithm is also introduced. Section VII summarizes the conclusions.

 ISBN: 978-1-61208-427-5

II. THE PROPOSED ALGORITH

The algorithm is a novel authenticated scheme that provides data confidentiality. By confidentiality we mean data encryption and data authentication / integrity. A secret key of 90 bits is used along with a randomly chosen number to encrypt the data. Range of the random number is 0 to 2^{90}-1. The random number is used with the secret key to generate subkeys for data blocks. The idea of generating these subkeys is as follows:

A group of nine coaxial disks are used to generate subkeys. These disks are represented in a physical form (for explanation purpose) in Fig. 1. Each disk is divided into 2^{10} slots. The slots are numbered 1, 2, 3, 1024; each slot carries a value 1, 2, to 1024. Initially, each slot carries a value equal its number e.g., slot number1 carries value 1, slot number 2 carries value 2 and so on., but after turning the disk with respect to a specific pointer P each slot carries a value that is different than its number e.g., If the disk is turned anticlockwise by 3 slots, then slot number 1 carries value 4, slot number 2 carries value 5 and so on, Fig. 2 and Fig. 3.

The secret key is composed of 9 blocks: S_1, to S_9, Each key block is 10 bit long. The value of any key block can be chosen as a pointer value P e.g., suppose the secret key is S= 1000 1022 0300 0400 0500 0250 1023 0450 0333. P value can be S_9 i.e., 333. Each data block has a corresponding disk, that is data block number 1 corresponds to disk number 1, data block number 21 corresponds to disk number 21 mode 9 = 3. Each disk turns by a specific value that is different from other disks, and the slots values corresponding to P on each disk (after turning) are used to generate the subkeys as shown hereinafter.

The proposed algorithm can be divided into the following steps:

A. Dividing the Message into Blocks

The message is divided into blocks of 10 bit each. B_1, B_2, B_n., number of blocks in the message is n.

Also, the secret key of 90 bits has 9 key block,
S_1, S_2, ……,S_9.

Each data block has also a corresponding key block.

B. Generating a Random Number

In every encryption process choose a random positive integer number R of any value less than (2^{90}-1).

C. Compute Hash Function

Groups of data blocks are formed from nine block, i.e., GB_1, GB2,……, GB_m. Then, XOR ing GB_1 (first group of blocks) with the second GB_2, and the result is XOR ed with the third group of blocks GB_3, and so on till the message is finished

$$MAC = GB_1 \oplus GB_2 \oplus GB_3 \ldots\ldots \oplus GB_{n.} \quad (1)$$

GB_1:

B_1	B_2	B_3	B4	B_5	B_6	B_7	B_8	B_9

GB_2:

B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}

GB_3:

B_{19}	B_{20}	B_{22}	B_{22}	B_{23}	B_{24}	B_{25}	B_{26}	B_{27}

D. Generation of Subkeys

Generation of subkeys is performed into two steps, computing the turning value of the disk that corresponds to the data block number, and then using the turning value with the pointer value to compute the subkey for this data block as follows:

a. Each data block has a non repetitive subkey which is different from all other subkeys. Random number R is used with the secret key to generate the subkeys for each data block. Number of subkeys to be generated equals number of data blocks n, i.e., for data block number (I) a corresponding disk J. Disk J turns by value TV_J:

$$TV_J = (I + R)^{S_J} \bmod (2^{10} - 1) \quad (2)$$

Where: J is disk number

I is data block number.

^{S_J} is secret key block number.

$$\left.\begin{array}{ll} J = I & \text{for } I \le 9 \\ J = I \bmod 9 & \text{for } I > 9. \end{array}\right\} \quad (3)$$

These two equations (2) and (3) produce different turning values for each disk in every encryption process. As a consequent, subkeys are not repeated, as an example: For

the secret key is : S = 1000 1022 0300 0400 0500 0250 1023 0450 0333. Let the random number is 123456789, and let the pointer value be the value of S_9, i.e., P = 0333.

Disk number 1 will turn anticlockwise by value TV_1:

$TV_1 = (1+123456789)^{1000}$ mode $(2^{10}-1) = 397$.

Disk number 2 will turn anticlockwise by value

$TV_2 = (2+123456789)^{1022}$ mode $(2^{10}-1) = 16$.

And so on.

b. The subkey for data block number I is computed according to the turned value of its corresponding disk, and is given by:

$$SK_I = (P + TV_J) \text{ mode } (2^{10}-1) \quad (4)$$

For block number 1, its subkey is:

$SK_I = 397 + 0333 = 730 = 10110\ 11010$.

Similarly, subkey for block number 2

$SK_2 = 16 + 0333 = 0349 = 01010\ 11101$

III. ENCRYPTION PROCESS

Given: plaintext B, secret key S, random number R, MAC, and computed subkeys SK_I.

We start by computing a function of R e.g., R~ using the secret key and the pointer value. Then we encrypt R~ by mutation and crossover with a fixed number X1. The same procedure is repeated with the MAC number except that we use R in computing the function of MAC e.g., MAC~.

Let the secret key S = S1 S2 S3 …..S9, and the plaintext divided into blocks e.g.,

B_1 = 1101101101, B_2=1011100101,
B_3 = 0101010111, B_4, ……,B_n.

A. Encrypt the Random Number R

In this regard, we use two fixed large numbers known in sending and receiving algorithms X_1, X_2. Also, we do not use R or MAC or S themselves, but functions of them.

$$ER = R^{\sim} (MuCr)\ X_1 \quad (5)$$

Where: MuCr stands for (Mutated and Crossed over) as explained for data blocks hereinafter.

$R^{\sim} = (R \pm X_2) \oplus SR$.

$SR_i = p^{10}$ mode $(S' \oplus i)$. i= 1, 2, ……9.

$S' = S_1 \oplus S_2 \oplus S_3 \oplus \ldots .. , S_9$

X_1, X_2 are two large numbers.

B. Encrypt Message Authentication Code MAC

$$EMAC = MAC^{\sim} (MuCr)\ X1 \quad (6)$$

$MAC^{\sim} = (MAC \pm X_2) \oplus SM$.

$SM_i = p^R$ mode $(S' \oplus i)$. i= 1, 2, ……9.

These two encrypted values ER and EMAC form the first 180 bits of the encrypted message. See Fig. 4.

C. Mutation:

For each block of data apply mutation process at a specific arbitrary bit number (say bit number 4):

B_1= 1101 ‖ 101101, B_2=1011 ‖ 100101 and

B_3 = 0101 ‖ 010111 would be :

B_1'= 1011011101, B_2' = 1001011011, B_3' = 0101110101.

D. Crossover:

At another arbitrary bit number (say bit number 6), B_1' will be crossed over with B_2', B_2' with B3', and so on till B_n' with B_1'as:

B_1' = 101101 ‖ 1101, B_2' = 100101 ‖ 1011,

B_3' = 010111 ‖ | 0101

B_1'' =101101<u>1011</u>, B_2'' = 100101<u>0101</u>

E. Blocks Encryption

Fig. 5 explains data blocks encryption. Each Block after mutation and crossover is encrypted by its subkey, so the encrypted block EB_I is computed as:

$$EB_I = EB_{(I-1)} \oplus B_I'' \oplus SK_I \quad (7)$$

 ISBN: 978-1-61208-427-5

e.g., $EB_1 = B_1" \oplus SK_1$

= 10110 11011 ⊕ 01110 01100 = 11000 10111.

Also, $EB_2 = EB_1 \oplus B_2" \oplus SK_2$.

= 11000 10111 ⊕ 10010 10101 ⊕ 01010 11101

= 00000 11111 ……..etc.

F. Summery

The algorithm block diagram is shown in Fig. 6.

A message of m bit, and a 90 bit security key $S = S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9$ can be encrypted as follows:

a- Choose a positive integer random number R;

$0 < R < 2^{90}$.

b- Divide the message into 10 bit blocks, each nine blocks form a group of 90 bit. Get the hash message authenticated code (MAC) by XOR ing groups as explained in (1).

c- Compute $R^{\sim}$ as function of R, and $MAC^{\sim}$ as a function of MAC. Then, encrypt $R^{\sim}$ and $MAC^{\sim}$ by XOR ing them with the X1, X 2 as explained in (5) and (6).

d- Generate a turning value for each disk as shown in (2) and (3).

e- Compute subkeys for each data block by (4) using the turning value of the corresponding desk.

f- Apply mutation and crossover operations on each block at an arbitrary bit number for each operation.

g- Encrypt each data block B_I of the information message using previous blocks and its subkey as in (7).

G. Decryption

The algorithm reads ER first to decrypt the random number R; $R= (R^{\sim} \pm X_2) \oplus S^{\sim}$. Getting R the algorithm decrypts EMAC to check the authenticity of the message (after being decrypted). Then the procedure goes on, in reverse order.

IV. EXAMPLE

Strict avalanche criteria(SAC) is a desirable property of cryptographic algorithms. That is if an input plaintext is changed slightly (e.g., flipping a single bit) the enciphered output changes significantly (may be more than half the output bits flip).

Example: The input plaintext is " DISASTER". Flipping one bit from the plaintext, we get "DISCSTER", (one flipping A (01000001) to C (01000011)). The Key used is “SRIRAMSR".

DISASTER encrypted message is 00111,11011 10001,10100 10101,01000 10100,10011 11011,01001 01001,11011 11011,00111

DISCSTER encrypted message is 01010,00110 00100,01011 01100,11100 11110,10000 00100,10111 01101,01110 11111,10010

Number of flipped bits in the encrypted message is 42 bit (out of 65 bits of the original message).

Avalanche effect = 42 * 100 / 65 = 64.6%.

The same example was carried out by other algorithms [24], and the results are shown in Table I.

TABLE I COMPARISON OF AVALANCH EFFECT

Encryption Technique	No. of flipped bits	%
Playfair Cipher	4	6.25
Vigenere Cipher	2	3.13
Caesar Cipher	1	1.56
DES	35	54.68
Blowfish	19	28.71
The Proposed Technique	42	64.6

V. ALGORITHM EVALUATION

A comparative evaluation of the algorithm is presented in this section. Two main properties are discussed here : performance, and randomness.

A. Performance

The algorithm runs on a 3.2 GH PC with different lengths messages, the concluded speed is 5.19 cycle per byte (587 MiB/s) for the encryption process, which is very fast compared to different algorithms shown in the Table II benchmark, [33].

 ISBN: 978-1-61208-427-5

TABLE II ALGORITHM SPEED COMPARISON

Algorithm	MiB/Second	Cycles Per Byte
AES/GCM (2K tables)	102	17.2
AES/GCM (64K tables)	108	16.1
AES/CCM	61	28.6
AES/EAX	61	28.8
CRC32	253	6.9
Adler32	920	1.9
MD5	255	6.8
DES/CTR	32	54.7
DES-XEX3/CTR	29	60.6
DES-EDE3/CTR	13	134.5
PROPOSED ALGORITHM	587	5.19

Number of overhead bits is fixed and irrelevant to the message length (double the key size) i.e., 180 bits only for MAC and random number).

B. Analysis

Although the algorithm does not depend on substitution permutation networks (SPNs), it keeps its cryptographic static and dynamic prosperities. The algorithm strict avalanche effect causes 65 % of bits in average to be flipped in the enciphered text for one bit flipped in plaintext as shown in the Table 1. Strict avalanche criterion is a measure of a cipher's randomness [23]. This ensures its resistance to statistical, clustering, linear, and differential cryptanalysis.

The algorithm provides randomized encryption, so that when encrypting the same message several times, it will produce different ciphertexts each time. Key size can be of any chosen number. Consequently, block size can be larger, e.g., key size of 1024 bit with block 128 or 256 or 512. The larger the key size the larger range for the random number. $0 < R < 2^{\text{key-size.}}$.

The algorithm has two different arbitrary bit numbers (from 1 to block_size -1) one for mutation and the other for crossover process.

Also, number of mutation then crossover rounds can be increased to any chosen number.

The number of brute force trials of an n bit message is: {TV = 9 (I) × 2^{90} (R) × 2^{90} (S) × 2 (Turning direction)} ×{SK= 2^{10} (P) }×{10^{10} (10 possible bit number for mutation, and 10 bit number for crossover for each one)}. This equals 2.82×10^{68} or 8.95×10^{51} years (assuming 10^{10} decryption process per second).

In a known message attack, if the attacker knows both plain and ciphered messages completely, he cannot get MAC or R. Both are not send, but functions of them. To get the function of R, i.e., $R^{\sim}$, the attacker has to make reverse mutation and crossover in 10^{10} operations, (nine possible mutation bit numbers, and for each one nine possible crossover bit number). While guessing R from $R^{\sim}$ and X_2, there are 2^{90} possible changes in $R^{\sim}$. So, the attacker needs $(10^{10})^{90}$ i.e., 3.1×10^{883} years (assuming 10^{10} operation per second). A similar number of operations to be executed to get MAC from MAC~.

The algorithm time and space complexity is O(n), i.e., the required time and space for encryption/decryption increase linearly with message length. However attacker algorithm is NP complete.

VI. CONCLUSIONS

In this paper, we introduced a novel immune scheme for block cipher randomized encryption. The scheme shows high strength of confidentiality even for known message attack. A hash message authentication code is also used to ensure message authenticity. In the proposed scheme, both the key length and the block size can be of any chosen size, provided key length modulo block size is zero.

The algorithm has a high degree of randomness; its strict avalanche effect is very significant and surpasses a lot of the famous algorithms. This proves its immunity to cryptanalysis. Knowing the algorithm, the plaintext and the ciphertext does not reveal useful information for the attacker to crack the key or the random number, since in every run a new random number and new subkeys are generated, consequently different ciphertexts for the same plaintext.

The algorithm has a strong strict avalanche criterion (SAC) (65% in average). Also, it keeps the cryptographic static properties of substitution permutation networks (SPNs) of completeness, nonlinearity. In the same time the algorithm provides perfect security defined by Shannon [22], and every bit in the information message is encrypted using a different subkey.

Implementing different random number of large range ($2^{\text{key\_size}}$) in every encryption process makes it impossible to be cracked. For the chosen length (90 bit), brute force attack needs 3. 7 X 10^{16} Years to crack that key, (assuming 10^{10} check per second).

The random number and MAC number are safely distributed between sender and receiver in a new procedure. The hash function used ensures the authentication and integrity of the message. Any bit change will be detected.

As an additional feature of the scheme some of its parameters are selective and not fixed: Key size – Block size – Bit number to perform mutation – Bit number to perform crossover – Disk number to be used as a pointer.

 ISBN: 978-1-61208-427-5

A good feature of the algorithm is that, its time and space complexity is O (n). So, the required memory resources and computation time are not increased in a large scale with the increase of message length. In the same time the attacking algorithm is NP complete.

REFRENCES

[1] D. Kahn, " The code breakers", ISBN 0-684-83130-9 New York, Macmillan ,1967,

[2] "Cryptology (definition)", http:// www.marriam-webster.com/dictionary/cryptology, Merriam-Webster's Collegiate Dictionary (11th edition Ed.).Merriam- dictionary / cryptology, retrieved 2015-03-25.

[3] H. Beker and F. Piper, "Cipher Systems: The Protection of Communications", John Wiley & Sons, New York, 1982.

[4] D. W. Davies and W.L. Price, " Security for Computer Networks", John Wiley & Sons, New York, 2nd edition, 1989.

[5] D.E. Denning, "Cryptography and Data Security", Addison-Wesley, Reading, Massachusetts, Reprinted with corrections1983.

[6] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22,644–654,1976.

[7] W. Diffie, "The first ten years of public key cryptology", G.J. Simmons, editor, Contemporary Cryptology: The Science of Information Integrity, 135–175, IEEE press,1992.

[8] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques", Proceedings of AFIPS National Computer Conference, 109–112, 1976.

[9] D. Kahn, "The Code breakers", Macmillan Publishing Company, New York, 1967.

[10] A.G. Konheim, "Cryptography, A Primer", John Wiley & Sons, New York, 1981.

[11] G. J. Simmons, editor, "Contemporary cryptology: An

introduction", Contemporary Cryptology: The Science of Information Integrity, 1–39, IEEE Press, 1992.

[12] R. Merkle, " Secrecy, Authentication, and Public Key Systems", UMI Research Press, Ann Arbor, Michigan, 1979.

[13] Boritz, J. "IS Practitioners' Views on Core Concepts of Information Integrity". International Journal of Accounting, Information Systems. Elsevier, August 2011.

[14] W. Diffie and M. Hellman, "New directions in cryptography". IEEE Trans. On Information Theory, IT-22(6), 1976.

[15] National_ Bureau_ of_ Standards," Data Encryption Standard (DES)," federal Information Processing Standard Publication FIPS PUB 46-3,U.S. dept. of commerce/national institute of standards and technology, 1977.

[16] I. Ben-Aroya and E. Biham, " Differential cryptanalysis of Lucifer". In D.R. Stinson, editor, Advances in Cryptology: CRYPTO'93, LNCS 773, 1993.

[17] D. Warren,"1. AES seems weak. 2. Linear time securecryptography",http://www.researchgate.net/publication/220335792_1._AES_seems_weak._2._Linear_time_secure_cryptography, IACR Cryptology ePrint Archive 01/ 2007, retrieved 2015-03-01.

[18] Bruce Schneier, "Another New AESAttack" https://www.schneier.com/blog/archives/2009/07/another_new_aes.html, retrieved 2015-03-03.

[19] E. Biham and A. Shamir, " Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, vol. 4, no. 1, pp 3-72,1991.

[20] Niels Ferguson e.al. "Improved Crypt analysis of Rijndael " https:// www.schneier.com/ paper-rijndael.pdf Counterpane Internet Security, Inc., 3031 Tisch Way Suite 100PE, San Jose,CA 95128, retrieved 2015-03-01.

[21] L. Brown, J. Pieprzyk, and J. Seberry, " LOKI - a cryptographic primitive for authentication and secrecy applications". In J. Seberry and J. Pieprzyk, editors, Advances in Cryptology: AusCrypt'90, LNCS 453,. Springer Verlag, 1990.

[22] C. E.Shannon, " Communication theory of secrecy systems", Bell System Technical J. 28, 656-715,1949.

[23] Heys and Tavers , "On Design of Secure Block Ciphers", Queen's 17 th symposium on Communications, Kingstone, Ontario, Canada, May 1994.

[24] Sriram Ramanujam and Marimuthu Karuppiah, "Designing an algorithm with high avalanche effect", IJCSNS International Journal of Computer Science and Network Security, VOL. 11 NO.1, January 2011.

 ISBN: 978-1-61208-427-5

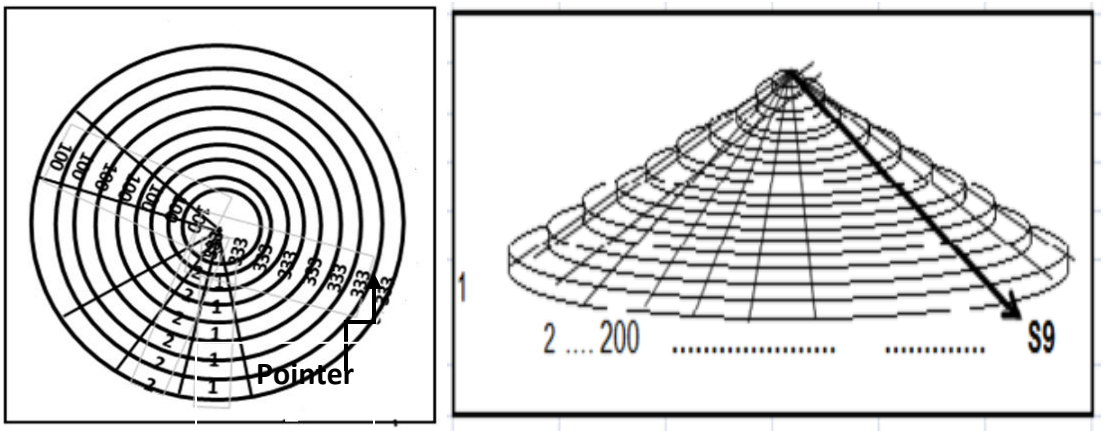


Figure 1 Physical Representation of Coaxial Disks

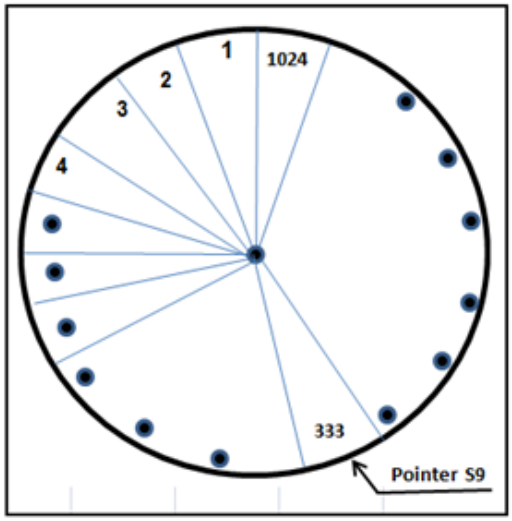


Figure 2 One Disk in Initial Position

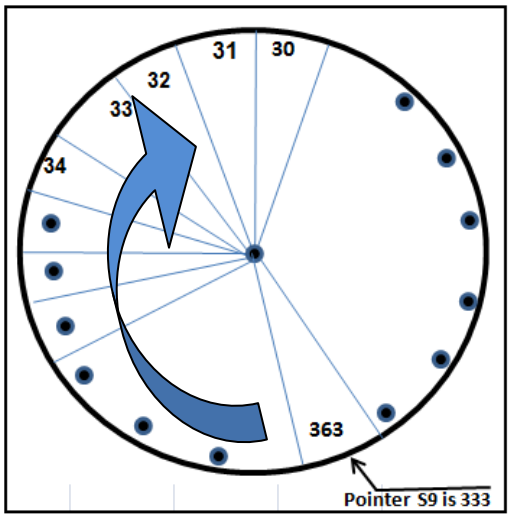


Figure 3 Same Disk after Turned 30 Slots

 ISBN: 978-1-61208-427-5

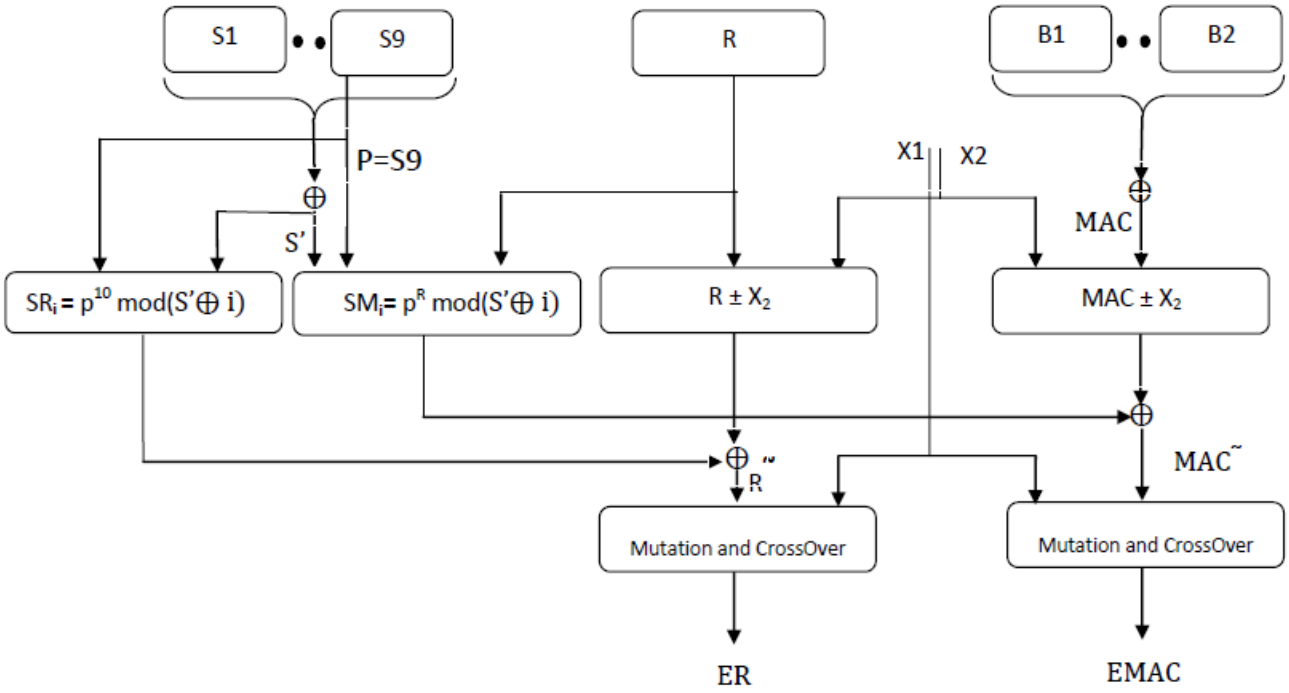


Figure 4 Encryption of Random and MAC Number.

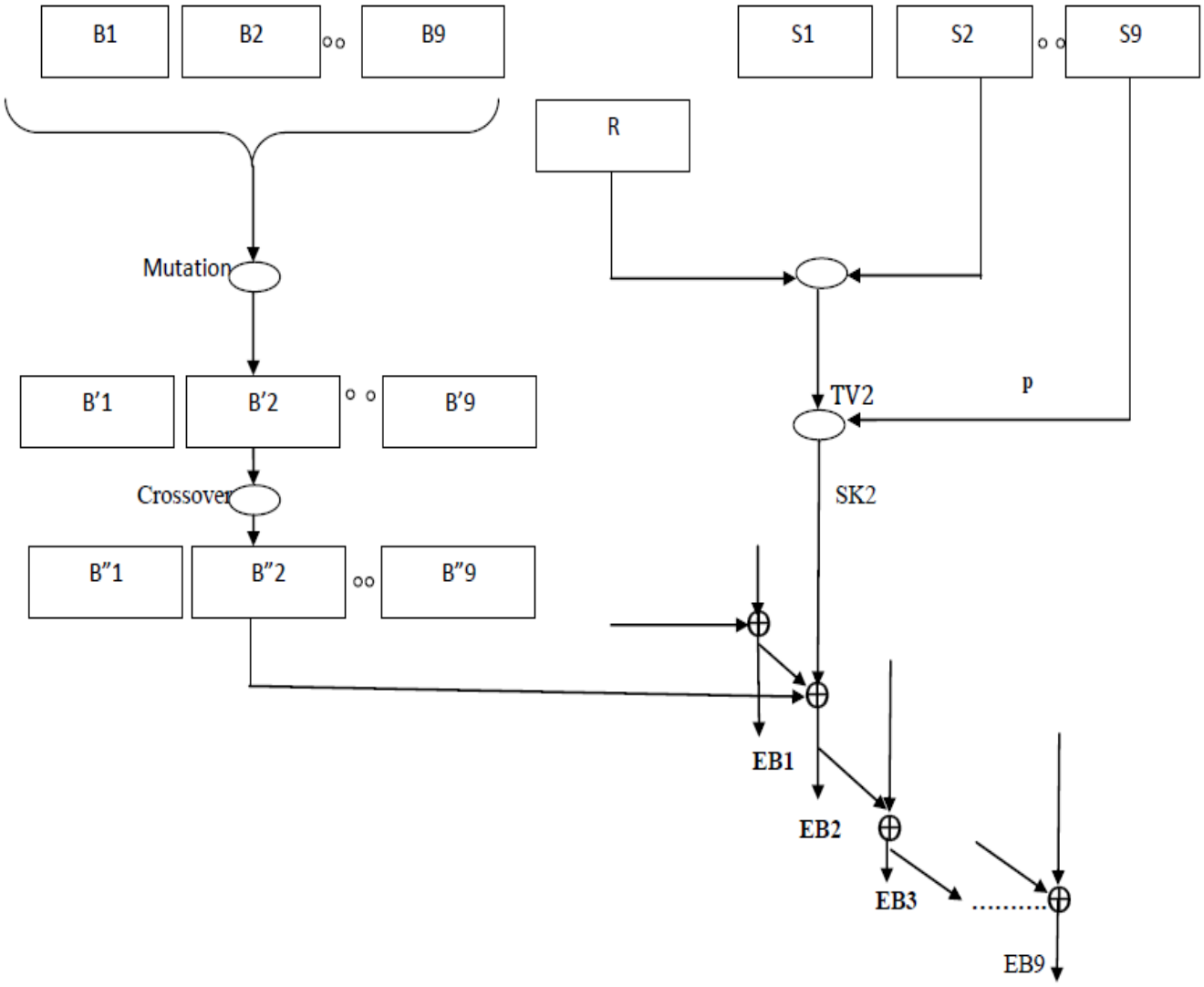


Figure 5 Encryption of Data Blocks

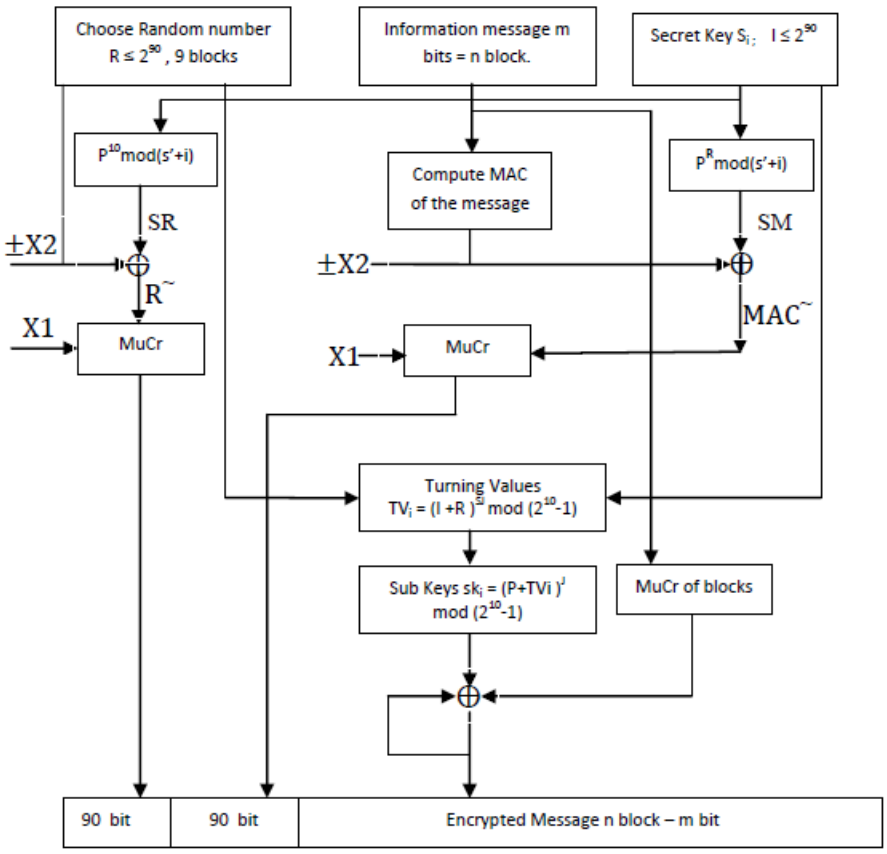


Figure 6 Encryption Scheme

New Directions in Applying Physical Unclonable Functions

Rainer Falk and Steffen Fries
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {rainer.falk|steffen.fries}@siemens.com

***Abstract*—Physical Unclonable Functions (PUF) realize the functionality of a "fingerprint" of a digital circuit. They can be used to authenticate devices without requiring a cryptographic authentication algorithm, or to determine a unique cryptographic key based on hardware-intrinsic, device-specific properties. It is also known to design PUF-based cryptographic protocols. This paper presents several new applications of PUFs. They can be used to check the integrity, or authenticity of presented data. A PUF can be used to build a digital tamper sensor. An identifying information in a communication protocol can be determined using a PUF, or a licensing mechanism can be realized.**

Keywords–physical unclonable function; key extraction; embedded security; licensing; configuration integrity

I. INTRODUCTION

The need for technical information technology (IT) security measures increases rapidly to protect products and solutions from manipulation and reverse engineering. Cryptographic IT security mechanisms have been known for many years, and are applied in smart devices (internet of things, industrial and energy automation). Such mechanisms target authentication, system and communication integrity and confidentiality of data in transit or at rest. One base for the operation of security mechanisms is typically a cryptographic key that has to be stored securely on devices. Upcoming industrial security standards (ISO/IEC 62443 [1]) require explicitly a hardware-bound storage for cryptographic keys.

A significant effort is often required in practical realizations to protect key storage, e.g., by external hardware integrated circuits (IC). In current research, IT security methods are investigated that directly use a unique physical property of an object as a physical fingerprint. Small random differences of physical properties are used to identify an object directly, or to derive a cryptographic key for conventional cryptographic IT security mechanisms [2]. A digital circuit, i.e., a semiconductor integrated circuit, can contain a digital circuit element called a physical unclonable function (PUF) to determine the physical device fingerprint. Minimal differences in the semiconductor structure, like for instance the doping of a semiconductor, the layer thickness, or the width of lines arise at the production randomly. This is similar to the random surface structure of paper sheets. These chip individual properties are "simply there" without being designed-in explicitly, or being programmed by a manufacturer during production. Such a device fingerprint is unique, and cannot practically be reproduced easily (unclonability). In addition, the fingerprint can be modified, or even destroyed when the IC is manipulated.

After giving an overview of some major realization possibilities for a digital PUF in Section II, basic usages of a PUF are summarized in section III. The main contribution of the paper is in section IV, describing several new applications of PUF technology. Section V concludes with a summary, and an outlook.

II. PHYSICAL UNCLONABLE FUNCTIONS AS DIGITAL DEVICE FINGERPRINT

A PUF can be realized on a semiconductor circuit to determine a device-specific piece of information depending on variations in the target physics due to the manufacturing process. The information provided by the PUF can be used directly for low-cost authentication, to determine a serial number as an identifier, or as cryptographic key. The semiconductor circuit can be an application-specific integrated circuit (ASIC), or a field-programmable gate array (FPGA). This section gives a short overview about PUFs. More detailed information is available in tutorials on PUFs [3][4][5][6].

PUFs have been a major topic of academic research. However, PUF technology is already applied commercially. Examples are Intrinsic ID [7], Verayo [8][9], Microsemi Smartfusion2 FPGAs [10], and NXP SmartMX2 [11].

Common digital circuits are designed to provide identical behavior on different ICs. However, a PUF circuit is designed to provide different results on different ICs, but identical or at least similar results on the same IC when the function is executed repeatedly.

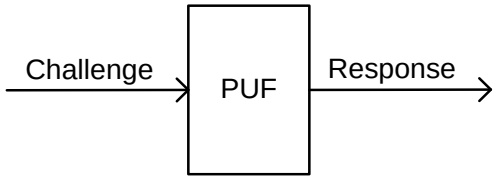


Figure 1. Challenge-Response-PUF

Figure 1 shows a challenge-response PUF, in which the PUF circuit determines a response value depending on a provided challenge value. Weak PUFs and strong PUFs are distinguished: while a strong PUF has a wide range of challenge input values, a weak PUF has no, or only a very

 ISBN: 978-1-61208-427-5

limited set of challenge values. A strong digital PUF can be realized by reconfiguring a digital PUF circuit depending on the challenge value.

The objective of a PUF circuit is that on the same IC, the response value for a given challenge value is stable (reproducibility), while on different ICs, the response values are different (uniqueness). As binary values are used for challenge and response values, the similarity can be measured by the Hamming weight, i.e., the number of different bits. The measure for reproducibility is the intra-device Hamming distance, i.e., the mean value of the number of different bits when the PUF is executed multiple times for a given challenge value. The measure for the uniqueness is the inter-device Hamming distance, i.e., the mean value of the number of different bits when executed in different ICs.

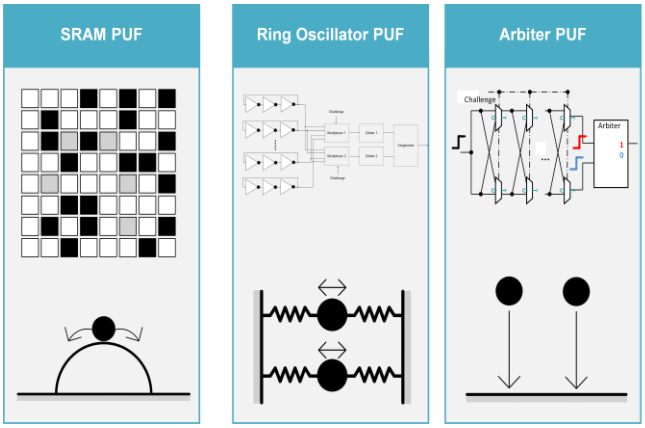


Figure 2. Example PUF Realizations and Their Mechanical Analogon

Figure 2 shows three examples of well-known constructions of PUFs and their mechanical analogon:

- SRAM-PUF: power-up value of static random-access memory (SRAM) cells
- RO-PUF (Ring oscillator PUF): oscillator frequency
- Arbiter PUF: time delay

Many more constructions for a digital PUF have been proposed, e.g., bi-stable Ring PUF, Flip-Flop-PUF, Glitch PUF, Cellular Non-linear Network PUF, or Butterfly-PUF.

A. *SRAM-PUF*

A digital memory can store binary values 0 and 1. After power-up, some memories show a device-specific initialization pattern. The power-up value of a memory cell can be either 0 or 1, or being instable (sometimes 0, sometimes 1). The pattern of power-up values of its memory cells is characteristic of a memory IC, depending on small variations of the semiconductor physics of each memory cell.

A mechanical analogon for the power-up is a ball placed on the top of a hill [12]. When the whole geometry is exactly symmetric, the ball will roll-down to the left side and to the right side with the same probability. If the hill, or the ball, would have some asymmetries from manufacturing, the ball will tend to roll-down either to the left side or to the right side.

B. *Ring Oscillator PUF (RO-PUF)*

A digital circuit can realize an oscillator using a delay circuit with a feedback loop (ring oscillator). The oscillation frequency depends on manufacturing variations. The frequency of two identically designed oscillators can be compared using a counter, and comparator. Depending on the IC, one or the other will oscillate with a higher frequency. Realizing multiple oscillators, a “fingerprint” of the digital circuit can be obtained.

A mechanical analogon is an oscillating mass, and spring. Two identical physical realizations will in practice have a slightly different oscillation frequency, depending on small physical variations.

C. *Arbiter PUF*

A further effect that can be used to build a PUF is time delay. Two identically designed signal paths will show minimal differences in the respective delay. After giving in input signal to both signal paths at the same time, an arbiter circuit determines the faster signal path, i.e., the signal path on which the signal appears first,

A mechanical analogon is a drop test for two identically manufactured masses. Depending on variations in the height, or the surface of the masses, one will tend to impact first on the floor.

III. BASIC PUF APPLICATIONS

A PUF can be used for security purposes in different ways. It can be used as low-cost object authentication, or to determine a cryptographic key. This section describes these two basic applications, and gives examples for some specific usages of PUFs.

A. *Object Authentication*

Authentication is an elementary security service proving that an entity in fact possesses a claimed identity. Often natural persons are authenticated. The basic approaches a person can use to prove a claimed identity are by something the person knows (e.g., a password), by showing something the person has (e.g., passport, authentication token, smart card), or by exposing a physical property the person has (biometric property, e.g., a fingerprint, voice, iris, or behavior). Considering the threat of counterfeited products (e.g., consumables, replacement parts) and the increasing importance of ubiquitous machine-based communication, also physical objects need to be authenticated in a secure way. Various different technologies are used to verify the authenticity of products, e.g., applying visible and hidden markers, using security labels (using, e.g., security ink or holograms), and by integrating cryptographic authentication functionality in wired product authentication tokens, or Radio Frequency Identification (RFID) authentication tags.

An object or digital circuit can be identified by a serial number. For authentication, a cryptographic authentication protocol can be used, requiring a secret/private key to be available on the object to be authenticated.

 ISBN: 978-1-61208-427-5

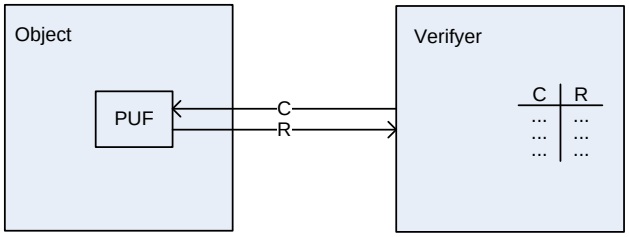


Figure 3. Challenge-Response-Authentisierung

For authentication, a challenge value is sent to the object to be authenticated. A corresponding response value is sent back and verified. The response is determined by the PUF. As only an original product can determine the correct response value corresponding to a challenge, the product entity or a dedicated part of the product is thereby authenticated.

Figure 3 shows how an object becomes authenticated by a verifier. The verifier maintains a database of reference challenge response pairs. For example, the database was filled during production of the object by recording arbitrary challenge-response-pairs. During the authentication the verifier selects a challenge value of the database and sends it to the object to be authenticated. The response value R is determined by means of the PUF, and transferred back to the verifier. The verifier compares the received response value with the reference value stored in the database. If these are similar, i.e., the number of different bits does not exceed a threshold, the object as authenticated successfully.

B. Cryptrographic Key Extaction

A cryptographic key can be determined based on inexact, noisy data. A "fuzzy key extractor" is a functionality that determines a stable cryptographic key using a PUF, and helper data [13][14]. The helper data allows to correct bit errors of responses (noisy data), and to map the PUF output to a given cryptographic key. A main advantage is that no secure non-volatile memory is needed on the device to store a cryptographic key.

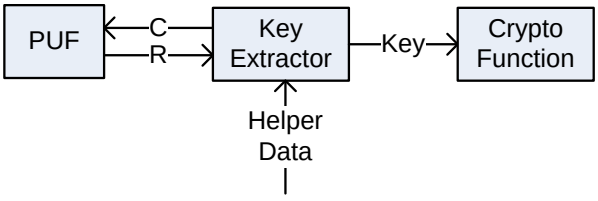


Figure 4. PUF Key Extraction

The PUF is used internally within a digital circuit to determine response values, see Figure 4. The helper data does not have to be stored securely. It can be used only by a single IC to determine the cryptographic key on the device.

C. Further PUF Applications

Several further applications, besides the two basic PUF usages stated above, have been proposed and designed. The following list section gives an overview on related work.

- A PUF can be used to prevent utilizing specific features of semiconductor ICs. Without chip-specific aiding information, the performance of an IC is reduced or access to certain memory partitions is prevented. Also, a PUF can be used to bind software intellectual property to a FPGA device by encrypting the software code using a PUF-generated device key [15], which is typically done during manufacturing. This solution can be used to protect for instance remote software updates [16].
- A PUF can be used to protect the execution of software code: the Control Flow Graph of an executed program depends on the output of a PUF [17].
- It is known to include a measurement value determined by a sensor as part of the challenge of a PUF to authenticate the sensor measurement [18][19]. This allows authenticating sensor measurements.
- A PUF can be used also in data communication to determine a message integrity checksum (message integrity code, message authentication code) [20]. While a real, physical PUF is used to determine the message authentication code by the sender, a simulated, algorithmic model of the PUF is used to verify the checksum by the receiver.
- Furthermore, the cryptographic key derived by a PUF of a semiconductor can be used to decrypt configuration data [21].
- A PUF can, as security primitive, be integrated in a cryptographic protocol directly [22][23].

D. Limitations of PUF

Building security solutions using PUFs, it is important to understand their limitations. Important issues to be considered are:

- Attacks on PUFs, and support functions as a fuzzy key extractor need to be taken into account. This relates for instance to the PUF model building, potential side channel attacks, and also fault injection attacks
- Robustness of a PUF implementations with respect to tamper resistance, e.g., how vulnerable is a solution with opened chip housing
- Reliability of the PUF with respect to the long term application in devices related to ageing, environmental conditions as temperature and others.
- Required processes for enrollment of data, which relates on one hand to the helper data on device, and within backend systems. On the other hand, depending on the PUF application the handling of the recorded challenge response pairs needs to be defined, as this information is sensitive and can be system critical. The latter may be compared to the handling of symmetric device keys, which have a similar level of sensitivity.

Based on these points it becomes even more obvious, that a security solution exposing the PUF functionality to other elements needs to designed PUF aware, especially

 ISBN: 978-1-61208-427-5

considering reliability and resilience requirements for long lasting deployments.

IV. New Applications of Phyiscal Unclonable Functions

Applications of PUFs fall basically in two categories: challenge response authentication, e.g., for low cost RFID Tags, and extraction of a symmetric cryptographic key. When used for protecting embedded systems, the cryptographic key can be used independent of PUF properties.

In this section, we describe potential new applications of PUFs in the context of security services.

A. Authentication Verification

It is known to use a PUF to authenticate an integrated circuit or a device respectively. However, the reverse is possible as well: the PUF can be used to verify the authentication of an external party. This approach has the clear advantage that no cryptographic algorithm has to be implemented to perform authentication checks. It rather requires the storage of a certain number of challenge-response pairs.

One application for this authentication verification can be, e.g., in the context access verification to a diagnosis or debug interface of an integrated circuit, or to protect the wake-up functionality offered by these chips.

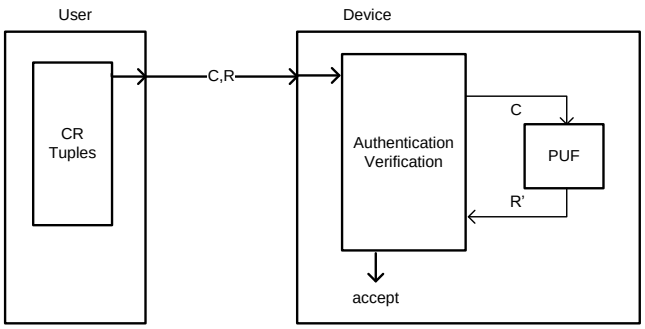


Figure 5. PUF Authentication Verification

Figure 5 shows how a PUF can be used to check authenticated access to a device or device functionality: a user presents a C/R pair of challenge C and response R. The PUF determines the response R' for the given challenge C. If the presented response R, and the determined response R' are identical or differ only in a limited number of bits, access is granted (accept). The C/R pair can be determined in different ways:

- In an initialization phase, C/R pairs can be read out from the device, and stored in a secure data base. Before the IC is put in operation, the interface to read out C/R pairs is blocked, e.g., by burning a security fuse.
- Should the PUF be a PUF for which an algorithmic model can be determined (as described in [20], the algorithmic model of the PUF can be used to compute C/R pairs.

B. Configuration Integrity Check

In a similar way, the integrity of configuration data can be verified by a PUF, directly.

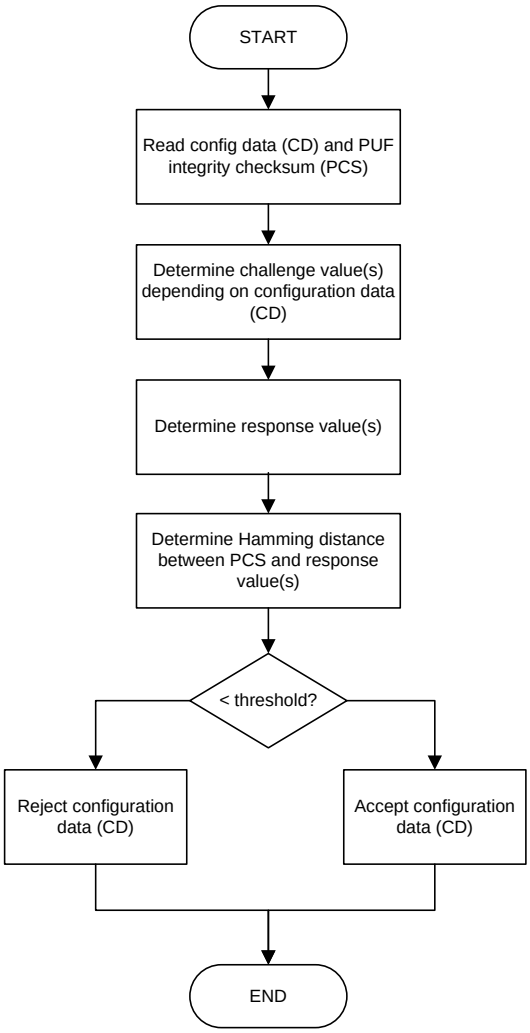


Figure 6. PUF-based Integrity Check of Configuration Data

Figure 6 shows a realization option: configuration data is read from an external, unprotected configuration memory, e.g., a serial electrically erasable programmable read only memory (EEPROM). Besides the configuration data (CD), a PUF checksum (PCS) is also read. A PUF challenge value is determined depending on the configuration data, e.g., a cryptographic hash value of the configuration data. The corresponding response value R' is determined, and the Hamming distance between R' and the PCS value is determined. The configuration data are accepted if the number of different bits is below a given threshold value.

A PUF may be used in a similar way as a key derivation function for a cryptographic key *K*. Depending on the cryptographic key *K*, challenge values are determined. The PUF response value(s) are used to determine a (derived) key.

 ISBN: 978-1-61208-427-5

C. PUF Tamper Sensor and PUF Built-In Self Test

Challenge response pairs of the PUF are typically stored as reference data. The integrated circuit uses the reference data to check whether the PUF is working correctly. This can be used for different purposes:

- A PUF-based tamper-sensor can be realized: when a tampering of the device occurred, the PUF provides different response value with high probability.
- A built-in self test functionality can be realized for a PUF, used, e.g., for authentication, or key extraction. Only if the PUF works as expected, the self-test succeeds.

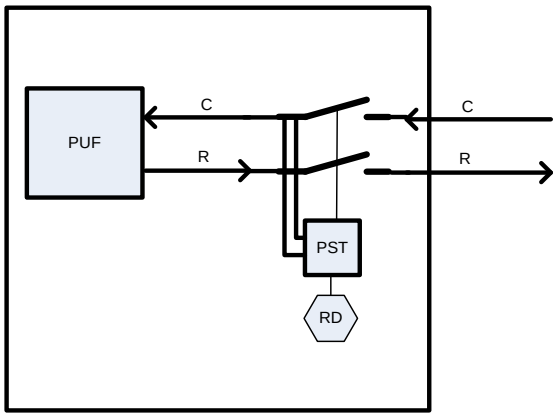


Figure 7. PUF Built-In Self Test

Figure 7 shows a realization option where reference data (RD) are used to check the PUF. Only if the PUF provides responses sufficiently similar to the reference data, access to the PUF is enabled by the PUF self-test unit PST.

D. Identifying Communication Sender

A PUF can be used to derive a serial number of a device. This PUF derived serial number or a derivation of thereof can be used to determine an identifier for data communication.

For example, an IPv6 stateless address auto configuration can be performed using a PUF. T. Aura defines how an IPv6 address can be created cryptographically [24]. Similarly, a PUF can be used to determine an IPv6 address. The challenge can be determined based on network part of the IPv6 address assigned by an IPv6 router. The host part is created depending on the PUF response output.

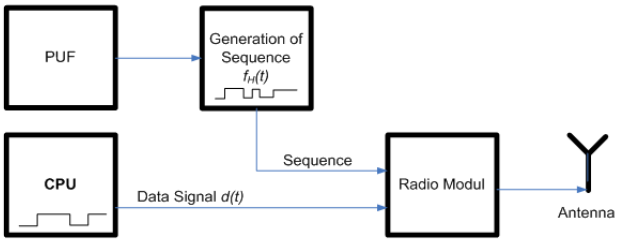


Figure 8. PUF-based Spread-spectrum Transmission

Figure 8 shows a different variant where the PUF-based identifying information is not included in the sender address. Instead, if a wireless spread spectrum transmission system is used, a spreading code is build or modified respectively depending on the PUF response. Hence, the PUF is used to realize a kind of "stream cipher" as spreading code.

E. PUF-helper Data as License File (license key)

A fuzzy key extractor allows determining a given cryptographic key using helper data. The helper data has two purposes: it allows correcting random errors of the PUF response, and it transforms the device-specific PUF response to a given cryptographic key. These properties can be used as licensing mechanism.

In a licensing scheme, a license code, or license key, is required to use a certain, software-based feature. The license code/key can be checked, resp. a key to decrypt code can be determined based on the license key.

With PUF helper data, the license code/key can be provided in the form of helper data: as long as the required helper data is not available, the license key cannot be built by a device. However, if helper data to reconstruct a certain license code/key is provided, the device can determine the license code/key. As a PUF is used, the helper data can be processed only on the single intended target device.

V. CONCLUSION

Physical unclonable functions have been investigated extensively by both research, and industry. The work focuses much on design constructions to realize a PUF, analyzing their statistical, and security properties, and on key extraction. Although being known for at least 10 years, one limited number of examples for commercial applications exists. Besides the classical usages, object authentication, and key extraction, a PUF can be specific new usages can be realized based on a PUF. This paper described several new applications for PUFs in different systems, either self-contained, like the tamper sensor or in conjunction with other parts of target solutions like in the case of licensing. These new applications are discussed as abstract concepts and need to be investigated and realized to gain more experience about the actual feasibility in products or solutions. This work is envisioned for the future.

Issues for the practical application are the stability over time (ageing), and under changing environmental conditions. As PUFs are still a relatively new security feature that is not yet broadly applied in practice, careful analysis of the actual security level as to be performed (e.g., modeling attacks, physical attacks, side channel attacks). The security management of PUF-based security solution has to be designed (e.g., enrollment of key material, building and maintaining databases comprising challenge/response pairs).

However, PUFs show unique properties that make them interesting for practical usage: they allow "storing" a cryptographic key in a protected way without requiring physical non-volatile memory. Low-cost authentication solutions can be built that do not require implementations of cryptographic algorithms.

 ISBN: 978-1-61208-427-5

REFERENCES

[1] ISO/IEC 62443, "Industrial Automation and Control System Security" (formerly ISA99), available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx, last access: January 2015

[2] B. Gassend, "Physical Random Functions", Masters Thesis, MIT, February, 2003, available from: http://csg.csail.mit.edu/pubs/memos/Memo-458/memo-458.pdf, last access: January 2015

[3] C. Herder, Y. Meng-Day, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial", Proceedings of the IEEE, Vol.: 102 No. 8, Aug. 2014, pp. 1126-1141, available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6823677, last access: January 2015

[4] S. Devadas, "Physical Unclonable Functions and Applications, Presentation Slides", available from: http://people.csail.mit.edu/rudolph/Teaching/Lectures/Security/Lecture-Security-PUFs-2.pdf, last access: January 2015

[5] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", DAC 2007, June 4-8, 2007, pp. 9-14 ACM, available from: http://www.verayo.com/pdf/2007_PUF_dac.pdf, last access: January 2015

[6] S. Katzenbeisser, Ü. Kocabas, V. Rožic, A. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon", IACR eprint 2012/557, Sep. 2012. Online]. Available from: https://eprint.iacr.org/2012/557.pdf, last access: January 2015

[7] Intrinsic ID Technology, available from: https://www.intrinsic-id.com/technology/, last access: January 2015

[8] Verayo, "Physical Unclonabel Functions (PUF)", available from: http://verayo.com/tech.php, last access: January 2015

[9] Verayo, "Introduction to Verayo", available from: http://www.rfidsecurityalliance.org/docs/Verayo_Introduction_RFIDSA_July_9_08.pdf , last access: January 2015

[10] Microsemi, "SmartFusion2", available from: http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2, last access: January 2015

[11] NXP, "NXP Strengthens SmartMX2 Security Chips with PUF Anti-Cloning Technology", February 2013, available from: http://www.nxp.com/news/press-releases/2013/02/nxp-strengthens-smartmx2-security-chips-with-puf-anti-cloning-technology.html, last access: January 2015

[12] C. Böhm and M. Hofer, "Physical Unclonable Functions in Theory and Practice", Springer, 2012

[13] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data", Eurocrypt 2004, LNCS 3027, Springer, 2004, pp. 523-540, available from: http://www.iacr.org/archive/eurocrypt2004/30270518/DRS-ec2004-final.pdf, last access: January 2015

[14] B. Škorić, P. Tuyls, and W. Ophey, "Robust key extraction from Physical Uncloneable Functions", Applied Cryptography and Network Security, LNCS 3531, Springer, 2005, pp 407-422, available from: http://members.home.nl/skoric/security/PUF_KeyExtraction.pdf) , last access: January 2015

[15] Y. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security", 16th USENIX Security Symposium, 2007, pp. 20:1-20:16, available from: http://www.usenix.org/event/sec07/tech/full_papers/alkabani/alkabani.pdf, last access: January 2015

[16] M. Gora, A. Maiti, and P. Schaumont, "A Flexible Design Flow for Software IP Binding in FPGA", IEEE Transactions on Industrial Informatics, vol. 6, issue 4, Nov. 2010, pp. 719-728

[17] R. Nithyanand and J. Solis, "Theoretical Analysis: Physical Unclonable Functions and the Software Protection Problem", IEEE Symposium on Security and Privacy Workshop, 2012, pp. 1-11 available from: http://www.ieee-security.org/TC/SPW2012/proceedings/4740a001.pdf, last access: February 2015

[18] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling Attacks on Physical Unclonable Functions", Proc. of the 17th ACM conference on Computer and communications security, 2010, pp. 237-249, , available from: http://people.idsia.ch/~juergen/attack2010puf.pdf, last access: January 2015

[19] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor Physical Unclonable Functions", IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), June 2010, pp. 112-117, available from: http://isis.poly.edu/~kurt/papers/sensorpuf.pdf, last access: January 2015

[20] W. Bares, S. Devadas, V. Khandelwal, Z. Paral, R. Sowell, and T. Zhou, "Soft message signing", patent application, WO2012154409, Nov. 2012

[21] S. Devadas and T. Ziola, "Securely field configurable device", patent application, US2010/0272255, Oct. 2010

[22] M. van Dijk and U. Rührmair, "Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results", Cryptology ePrint Archive: Report 2012/228, April 2012, available from: https://eprint.iacr.org/2012/228.pdf, last access: January 2015

[23] M. Majzoobi, M. Rostami, F. Koushanfar, D. Wallach, and S. Devadas, "Slender PUF Protocol: A lightweight, robust, and secure authentication by substring matching", IEEE CS Security and Privacy Workshop, 2012, pp. 33-44, available from: http://www.ieee-security.org/TC/SPW2012/proceedings/4740a033.pdf, last access: March 2015

[24] T. Aura, "Cryptographically Generated Addresses (CGA)", RFC3972, March 2005, available from: https://www.ietf.org/rfc/rfc3972.txt, last access: January 2015

 ISBN: 978-1-61208-427-5

The Random Gate Principle

Sheagan John
Department of Mathematics
The University of the West Indies
Mona, Kingston 7, Jamaica
Email: sheagan.john@mymona.uwi.edu

Curtis Busby-Earle
Department of Computing
The University of the West Indies
Mona, Kingston 7, Jamaica
curtis.busbyearle@uwimona.edu.jm

***Abstract*---We present the main theoretical ideas behind a proposed symmetric key algorithm. We show that it can be fairly easily constructed from mathematical pseudo-random parameters and known secure cryptographic functions. We describe how time intervals can be used to establish our algorithm for encryption purposes. We will briefly discuss the decryption of messages passed through the algorithm.**

Keywords-algorithm; encryption; gate; symmetric

I. INTRODUCTION

There exists a plethora of encryption cryptosystems and algorithms of varying security. The 3DES [1] and AES [2] algorithms are well known examples of such symmetric key methods. In this paper, we give the theoretical construction for a symmetric key algorithm which we call the Random Gate Principle (RGP). A schematic overlay of the basic nature of the system is shown in Fig 1.

The Random Gate Principle is a symmetric key algorithm which relies fundamentally on the properties of wide ranging time intervals in order to ensure cryptographic security. The RGP, though unrelated to the public-key system underlying the concept of Merkle's puzzles [3], is influenced by them. While Merkle's puzzles use a large number of messages to hide a particular one, the RGP conceals this message through inserting "garbage" of arbitrary length within the original message itself. The algorithm separates a plaintext message into blocks of predetermined bit lengths and feeds each of these individual blocks through a set of logical gates, which allow passage of a given block after a randomly determined time period. In this way, during the period between the passage of successive blocks, a string of bits can be inserted, where the insertion length varies according to the time interval. As a result, the original message is hidden within a longer garbled one which is then further encrypted using two internally generated keys. The final output from the procedure has the property that no attacker can determine the length of the original message from that of the encrypted one, or from the length of time taken for encryption.

The main idea with regards to proposing the RGP is the ability to combine low complexity cryptosystems to create a secure encryption algorithm. The remainder of this paper is organized as follows. In Section II we provide an outline of the encryption procedure, in Section III some bounds related to the message length are calculated, in Section IV we give an example of some simple attacks against the algorithm, and in Section V the method of decryption is outlined.

Throughout the paper it is to be understood that the output of all deterministic functions changes with the master key used for the encryption process.

II. OUTLINE OF ENCRYPTION

A particular gate of the RGP is denoted by a_k or b_k as shown in Fig 1. The gates are mathematical constructs, essentially consisting of functions which either have a defined image for a given input or do not. These functions have a predetermined and static set of parameters which describe their restricted ranges and as such the set of valid inputs. The most important criterion is that no two gates may ever both consider the same input as valid. Each gate also contains a check function and this will be discussed in relation to A_{in} and A_{out}. The total number of gates, N, may be some predetermined value which is known by both the sender and receiver of the encrypted message. Alternatively, this value may vary accordingly with the output from some deterministic function known to both parties.

The space A_{out} contains a check function and two pseudo-random number generators (PRGs) which we shall denote as PRG_1 and PRG_2. We denote by A_{in}, the representation of a space for implementation of three main functions.

The first function separates the initial plaintext message of length L into individual blocks each of length $\frac{L}{N}$ with the ability to pad a string of zeroes to the last piece until it is of exact length $\frac{L}{N}$. Each block then undergoes a left circular shift, where the number of bit positions shifted is equal to the block's position within A_{in}. This shift is to prevent the blocks of a random

 ISBN: 978-1-61208-427-5

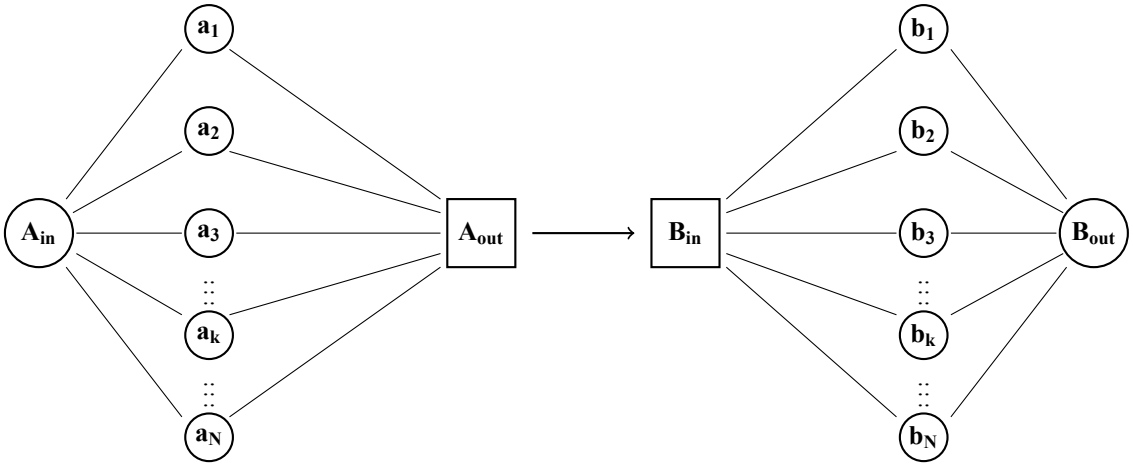


Fig. 1. A schematic presentation of the RGP gates.

message from exhibiting exceptional structure within the "garbage" surrounding them.

The first permuted $\frac{L}{N}$ block is prepended with the 16-bit representation of 1 and likewise each subsequent block till the last is prepended with that of the 16-bit representation of N. This concatenation will be denoted by $m||\frac{L}{N}$ where m is m'th integer.

The second function $H(X)$ takes each block of the form $m||\frac{L}{N}$, in order, and maps it to a gate a_k. This is not based on the parameters which govern the distinct valid inputs for a_k but rather on a conditional statement which assigns the prepended 16-bit value to a local variable within a_k if and only if a certain condition is met. In order for the choice of gate to be indistinguishable from random it is thus needed that the output of $H(X)$ also be indistinguishable from random. As an example, suppose that the Yarrow-160 [4] protocol is implemented within $H(X)$ in order to generate a random number, the last 32 bits of which are evaluated $mod(N+1)$. If the result is zero, a new number is generated and evaluated, otherwise the block $m||\frac{L}{N}$ is sent to the gate which has the number which is equal to the $H(X)$ output. In this manner the mapping $H(X) : A_{in} \longrightarrow a_k$ is independent of the message itself.

The third function is a check function which takes as input a number and outputs either 0, 1 or $\perp$(null). A gate, in order that only the blocks of the original message pass through it, does not assign a value to the local variable until the block is mapped directly to it from $H(X)$. Once the block is accepted, a single, static 256 bit number unique to each gate is relayed to the A_{in} check function. If this number matches the stored value associated with that gate, the check function outputs 0 and no new strings are fed into the $H(X)$ function. Simultaneously, a unique but static 256 bit number is sent from the gate to the check function of A_{out}, which outputs 1, thus initiating the two pseudo-random generators within A_{out}.

Each output of PRG_1 is a 32-bit number sent to and evaluated by all gates. The PRGs output cycle is attached to a counter which increases with every output and beginning with a value of 1 the counter value c increases to a value of N at which point the counter resets to 1. PRG_2 outputs a 16-bit number between 1 and N inclusive and prepends this number r to the output of the first PRG. The gate which receives this concatenated string as input and considers it valid, uses its check function to compare the r value to the value of the number m present in the $m||\frac{L}{N}$ block from $H(X)$. If the gate contains no information or the m value present is not equal to r, the check function will output 0 and upon receiving a zero value the check function within A_{out} outputs $\perp$ forcing PRG_2 to output 0 until the counter value c has reseted to 1. At this point PRG_2 will again output a valid non-zero number and prepends this to a new output from PRG_1 which the accepting gate again checks for equality to m.

If the two values are equal the check function of the gate outputs a single, unique, static 256 bit number to A_{out} which causes the check function of A_{out} to output 0 and terminate the loop for both PRGs. Simultaneously a unique 256 bit number is sent to the check function of A_{in} which outputs 1, enabling a new block to be fed into $H(X)$. As soon as the m value is checked to be equal to r the gate passes the reference value of its local variable to an array in A_{out} and is dereferenced in order for the variable to take a new value.

A. Bit Insertion

The data in A_{out} is sent through a one way route to B_{in} (see Fig 1) where the configuration of B_{in} is the same as that of A_{in}. Because of this, the complexity of implementation may be simplified by using the same

 ISBN: 978-1-61208-427-5

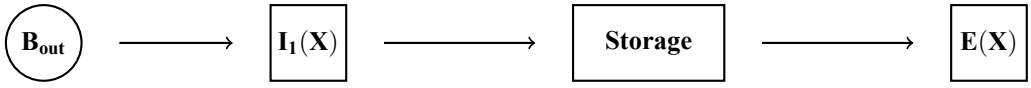


Fig. 2. The encryption process in stages.

set of functions and conditions within the gates for each pair, a_k and b_k. In fact the symmetry of the two halves can be made exact by using Yarrow-160 for both sets of two PRGs (PRG_1, PRG_2) and (PRG_3, PRG_4) present in A_{out} and B_{out}.

We denote by $I_1(X)$ the insertion function containing a one bit PRG linked to two counters. The blocks leaving B_{out} are once again in the form of $\frac{L}{N}$, having had the prepended 16-bit representation of m removed. Upon leaving, the first counter analyzes the constant data stream one bit at a time looking for an null space. Thus, upon finding the end of the first block, the counter increments from 0 to 1 while the PRG inserts a bit into each null space and does so until the counter detects the beginning of the second block. Since the PRG is only activated by the presence of an null space it will not replace any part of the original message. When each new block is detected, the counter increases and the PRG begins insertion once more. This cycle continues until the counter reaches N and resets, thus deactivating the PRG. The second counter records the number of bits $(B_1, B_2, \ldots, B_{N-1})$ inserted by $I_1(X)$ during each cycle.

The entire string is stored in a dynamic array called $Storage$ which holds each incoming bit in sequence. The insertion length values are similarly stored in separate arrays. Two functions are contained within $Storage$.

$D(X)$ takes as input the values $(B_1, B_2, \ldots, B_{N-1})$ and outputs $(C_0, \ldots, C_{N-1})$.

An insertion function, $I_2(X)$ uses a PRG- such as the Yarrow-160- to generate multiple outputs of random bits. These outputs are concatenated until they exceed the length of some value based on the $N-1$ insertion lengths. For example, a naive value may be given by

$$V_1 = \left\lceil \frac{B_1 + B_2 + \cdots + B_{N-1}}{|B_i - B_j|} \right\rceil \tag{1}$$

where the choice of i and j depends on the master key. At this point the concatenated string has all rightmost bits removed until it is of the exact length as V_1. The shortened concatenation is prepended to the garbage filled message . An similar procedure is followed for a second set of outputs of final length V_2, which is appended instead.

In the construction below, we denote the following: P, Q are large prime numbers which are known beforehand by both parties; $\alpha_i \neq j_i$ are numbers ranging from 1 to $N-1$.

$$C_0 = P(B_{j_1})^Q \tag{2}$$

$$C_1 \equiv B_{\alpha_1} (\mathrm{mod} B_{j_1}) \tag{3}$$

$$C_2 \equiv B_{\alpha_2} (\mathrm{mod} B_{j_2}) + C_1 \tag{4}$$

$$\vdots$$

$$C_{N-1} \equiv B_{\alpha_{N-1}} (\mathrm{mod} B_{j_{N-1}}) + C_{N-2} \tag{5}$$

Since each α_i, j_i is unique then each of the insertion bit values is used twice; once as a modular base and once as the number being reduced. The function $D(X)$ is deterministic, in that the sequence of pairs (α_i, j_i) is fixed regardless of the message being encrypted. This means that the value $B_{j_1} = B_k$ $(1 \leq k \leq N-1)$ for a fixed value of k. The importance of the equations given above is shown in Section V, with regards to decryption.

B. Keys and Authentication Codes

At this point the garbled message string and the set of values $C_0, C_1, \ldots, C_{N-1}$ are transfered to arrays in the space denoted by $E(X)$. This space contains an exclusive bitwise addition function and two deterministic functions, $E_1(X)$ and $E_2(X)$. The deterministic functions may be of varying mathematical complexity or may even be physically determined such as described in [6]. We do require that they be difficult to invert.

The purpose of $E_2(X)$ is to output two strings of 128 bits in length where the first output is prepended to the data and a second output is appended to the data. The first output of $E_2(X)$ may be determined by the length of the garbage filled message. The second output is determined using the value of the first output and thus by association would also be dependent on message length. The reliance on message length for generating input values for $E_2(X)$ underlines the importance of widely varying lengths of inserted bits.

Given a large range of possibilities, and since it is likely that any change in the first output will affect the second we can use these pairs as message authentication codes (MAC) [9]. See Section IV for more detail on this procedure.

We wish that an active attacker who either deletes bits from the encrypted message or inserts bits will, through tampering with message length, almost certainly invalidate one or both $E_2(X)$ outputs upon decryption. This is a useful implementation due to the simplicity of verification and given that MAC

 ISBN: 978-1-61208-427-5

authentication has been proven to be very secure [7]. To rigourously prove the claim of invalidation of $E_2(X)$ outputs would require a detailed explanation of how $E_2(X)$ works. Due to constraints on the amount of material which can be presented, we have decided to discuss this in future work.

Unfortunately we do not obtain non-repudiation [8] nor defend against an active attacker who does not change message length. This last concern can be somewhat countered by structuring $E_2(X)$ to depend not entirely on message length but on some other unique property such as a combination of message length and number of 0's contained in the message.

Two fixed length keys are outputted by $E_1(X)$ given the B_{j_1} value mentioned above as its input.

As an example, take a set of 512 bit keys, where the bitwise XOR function applies the first 512 bit key to the first 512 bits of the data. The next 512 bits of data (if more than 512 bits remains) is similarly XORed by the same key until all data is encrypted. This newly encrypted message is again XORed, but with the second key. It is after this procedure that the MACs are concatenated to the message.

III. MESSAGE LENGTH BOUNDS

In this section we must stress that the very nature of the randomness of the gate mechanism only allows for the calculation of *expected* values for most of the bounds. In particular there is no true upper bound as a gate may theoretically never be opened. In fact it is of great importance that the number of gates is small enough so as to allow a high *expectation* that every gate will open within some reasonable time span.

All the values obtained assume perfectly random behaviour. In practice there will likely be some small bias for particular gates.

Let us begin by calculating the lower bounds of time interval between exactly two consecutive blocks of form $m||\frac{L}{N}$. The denotation L is the length of the original plaintext, and $\bar{L}$ is the total length after insertions.

The lower bound can be found exactly, where $\mathbb{P}(X \to a_k)$ describes the probability of a block being sent to a given gate. Here, X represents a plaintext block. Assuming the Yarrow-160 protocol is indistinguishable from true random, then $\mathbb{P}(X \to a_k)$ will be $\frac{1}{N}$ whilst the time (t_h) to execute is constant regardless of the chosen gate. The probability $\mathbb{P}(X \to A_{out})$ is $\frac{1}{N^2}$ since the probabilities that the non-empty gate accepts the 32-bit number generated PRG_2 and the correct output from PRG_1 is prepended both have probability $\frac{1}{N}$. Assuming that both PRGs output the correct values then the time taken is just the constant $t_{g(o)}$ which is the time taken to open the gate and is a measure of the time taken for one complete cycle of the mechanism process described above. At this point the second block trails the first by a value of $t_{g(o)} + t_h$ but if the first block fails to pass through the second set of gates in one try then the second block will join it in B_{in}. Furthermore, if the second passes through immediately then the time interval remains $t_{g(o)} + t_h$. In fact this true for any two blocks for which the first block takes longer to pass through the second set of gates than the second block takes to pass through the first set of gates. It should also be noted that the time taken to send data from A_{out} to B_{in} is assumed to be negligible compared to $t_{g(o)}+t_h$ thus this value is what the lower bound is expressed as between consecutive message blocks.

The upper bound occurs when the time interval between two consecutive message blocks is greatest. This will occur when the first block passes through both sets of gates immediately whilst the second takes all tries. Using the probabilities above, we can show that the time taken for first block to pass from, A_{in} to B_{out} is equivalent to

$$T_1 = 2(t_{g(o)} + t_h) \tag{6}$$

The time taken for the second block to pass from A_{in} to A_{out} is

$$T_{2a} = (t_h + (N^3 - N)t_{g(r)} + t_{g(o)}) \tag{7}$$

where $t_{g(r)}$ represents the time taken for the gate to receive and reject r.

Since the probability $\mathbb{P}(X \to A_{out})$ is $\frac{1}{N^2}$ this means that there will be $(N^2 - 1)$ attempts where PRG_2 is forced to output 0 until the counter resets. Each counter cycle takes time $Nt_{g(r)}$ and thus for all incorrect attempts the total time is

$$(N^2 - 1) * Nt_{g(r)} \tag{8}$$

with the single correct attempt being of time $t_{g(o)}$. By the same argument the time

$$T_{2b} = (t_h + (N^3 - N)t_{g(r)} + t_{g(o)}) \tag{9}$$

which is the time taken for the second block to pass from B_{in} to B_{out} is the probabilistic upper bound. Therefore the total time interval is established as

$$2(t_h + (N^3 - N)t_{g(r)} + t_{g(o)}) - 2(t_h + t_{g(o)}) \tag{10}$$

assuming a perfectly random mechanism.

The effect of this time interval is to allow the one bit PRG within the insertion function to insert zeros and ones between two consecutive blocks. If the PRG performs an insertion every t_{ins} then the number of bits inserted between the two blocks has bounds given by equation (11).

 ISBN: 978-1-61208-427-5

$$\frac{t_{g(o)}+t_h}{t_{ins}} \leq n(Bits) \leq \frac{2[t_h+t_{g(r)}(N^3-N)+t_{g(o)}]-2(t_{g(o)}+t_h)}{t_{ins}} = \frac{2(N^3-N)t_{g(r)}}{t_{ins}} \quad (11)$$

This upper bound, however, is true only for the first two blocks after which the bound is represented by that of equation (12).

$$\frac{t_{g(o)}+t_h}{t_{ins}} \leq n(Bits) \leq \frac{(t_h+(N^3-N)t_{g(r)}+t_{g(o)})-2(t_{g(o)}+t_h)}{t_{ins}} = \frac{t_{g(r)}(N^3-N)-(t_{g(o)}+t_h)}{t_{ins}} \quad (12)$$

This wide ranging insertion length is meant to ensure that accurate prediction of the final length is extremely difficult. The probability of an inserted string being of a given possible length is $\frac{1}{N^2}$ and that of each $N-1$ insertion all having a particular length such that the sum of the lengths of all inserted strings is a given value can be approximated.

The probability of a message of original length L being mapped into that of maximum or minimum length; max($\bar{L}$), min($\bar{L}$) is

$$\frac{1}{N^2 * N^{N-1}} \quad (13)$$

since there is only one way of summing to either the maximum or minimum possible final length.

The probability $\mathbb{P}(L \rightarrow \bar{L})$ where the insertion length is non(max, min) increases with N and is largest for a total insertion length corresponding to N gate rechecks. We will determine the least number of repeats of the same message that may be processed before the output lengths are forced to match.

This can be solved exactly by considering the weak partition, $wp(N)_{N-1}$, of N into $N-1$ non-negative integers.

$$wp(N)_{N-1} = \frac{1}{(N-2)!}\prod_{i=1}^{N-2}(N+i) \quad (14)$$

$$P(L \rightarrow \bar{L}) = \frac{wp(N)_{N-1}}{N^2 * N^{N-1}} \quad (15)$$

Thus the number of repeated cycles of the same message must be no more than this absolute smallest value for mandatory correlation for any message length. The more numerous the gates, the larger this value becomes. At $N = 16$ this value is

$$\mathbb{P}(L \rightarrow \bar{L}) = \frac{wp(16)_{15}}{16^2 * 16^{15}} \cong \frac{1}{2^{40.89}} \quad (16)$$

and this represents the total insertion length most likely to be present. An equal total length, however does not indicate that the sequence of insertion lengths is the same, as the probability of a given inserted length is still $\frac{1}{N^2}$.

The last bound of importance is that of message length to number of gates. Since the security is based on random insertion length, the original message length must be at least N-bits. The upper bound of message length is determined by the security of encrypting a string of zeros, as will be shown in the next section.

IV. SIMPLE ATTACKS

We illustrate a chosen-plaintext attack. Consider an attacker who knows that the MAC lengths are 128 bits. We show the method by which an attacker can break the security in the shortest possible time with only this knowledge.

Assume the attacker sends a string of zeroes through the RGP to be encrypted. The garbled message has been XORed by two 512 bit keys, as in Fig. 3. Note that the circular shift on each block of zeroes does not affect it at all.

We let the length of $1st\,Division$, denoted by C, be arbitrarily long and thus the last $512 - V_1$ bits of the XORed keys will always be XORed with zeros. Knowing the combined length of the first MAC and B is $128 + V_1$ bits, the attacker assumes C is XORed with some $512 - V_1$ bit portion of the keys. The attacker knows that the encrypted $512 - V_1$ bits they are searching are in fact the keys themselves but does not know what V_1 is, and will not know even if the length of C is exactly 512 bits. This is because the beginning V_1 bits of the first 512 bit encrypted portion are different than the first V_1 bits of the subsequent portion and thus no repeating pattern yet emerges. As C exceeds 512 bits the sequence of the further bits exactly repeats that of the aforementioned $512 - V_1$ bits. At $C = 1024 - V_1$ bits the entire repetition is shown and the attacker can determine what the XOR value of the last $512 - V_1$ bits of the two keys is. This gained information is enough to determine which portions of the ciphertext are simply a long string of zeroes. The same method can be used if C is just a string of ones.

For complete security of any message encrypted with a key of length K it is thus recommended that the message be no longer than KN bits in length where N is the number of gates. This is easily remedied by forcing an initial message to be split into pieces and each piece fed into A_{in} in sequence, where new keys are generated each time. Another, less simple countermeasure is to construct $E_1(X)$ such that key length is dynamic and the keys may have differing lengths. Under these two improvements, chosen-

 ISBN: 978-1-61208-427-5

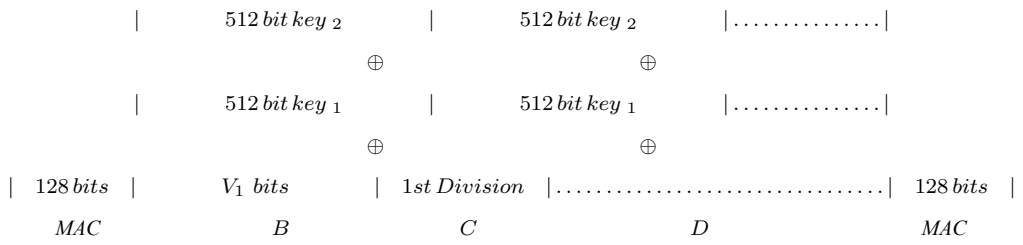


Fig. 3. Ciphertext output for an arbitrary message.

plaintext attack is not enough to break security. This is since the MAC, the value of V_1, the two keys, and the garbage inserted, constantly change with each encryption, even with the same plaintext used multiple times.

We will now allow the same attacker to record the time intervals between several different length message inputs and their encrypted outputs. We must show that such a timing attack is insufficient to determine with non-negligible probability which encrypted message the attacker is viewing. The attacker is given the power to submit an arbitrary length string of zeros or string of ones, both of which have length less than or equal to KN. The attacker then outputs the probability that the encrypted message is that of the zeros. The probability will be $\frac{1}{2} + \varepsilon$ where the value of ε must be non-negligible for the algorithm to be considered secure.

One way to accomplish this is by forcing all encrypted messages to remain in the system until a minimal time

$$2(N^3 - N)t_{g(r)} * (N - 1) * \alpha \qquad (17)$$

has elapsed, where α may be arbitrarily large. Though, theoretically, a message portion may take an infinite amount of time to pass through the gates, by choosing a large value for α the majority of message encryption times can be standardized. Such an α would depend on the number of gates and would be determined experimentally. The obvious downside to this approach is the artificially long encryption time for some messages.

A. MAC and Key Integrity

With regards to an attack on message integrity we consider an attacker attempting an existential forgery under a chosen-plaintext attack. Also the attacker wants to be able to create a message which has the same pair of MACs as a valid message.

Suppose that the attacker intercepts an encrypted message and attempts to use the same MACs on an arbitrary plaintext of his choosing. As was shown in Section III, even with only sixteen gates, the number of messages that must be encrypted to guaranteed message length collision is large. Even then, the probability that the number of 0's occurring is the same in both cases is unlikely. Given that the $E_2(X)$ outputs depend on these two factors, a breach of integrity reduces to finding a collision for $E_2(X)$. It is thus extremely important that $E_2(X)$ be sensitive to even very minor changes in its inputs as well as possessing a large range of outputs. Taking into consideration a birthday attack, the number of random messages an attacker must send before being guaranteed a collision pair for both MACs is

$$2^{64} \cdot 2^{64} = 2^{128}$$

This assumes that there is no exploitable bias in the construction of $E_2(X)$.

Similarly, trying to force decryption of an arbitrary message with an externally generated message authentication code is an undertaking by brute force.

Assuming the bounds of message length for security against pattern recognition are adhered to and the function $E_1(X)$ which generates the keys is not known, then any simple attack on the message through key integrity which does not rely on exploiting some unforeseen weakness in the $E_1(X)$ function will be a brute force attack on the key length.

V. MESSAGE RECOVERY

Consider a fully encrypted message which has been sent to a recipient along with the values, $C_0, C_1, \ldots, C_{N-1}$. The intended recipient must possess knowledge of various portions of the encryption process. Namely, the number of gates, the values of the prime pair P, Q as well as the sequence of pairs (α_i, j_i) (see Section II.A) must all be shared knowledge. It is given that the recipient's RGP decryption algorithm contains exact replicas of $E_1(X)$, $E_2(X)$, the circular shift function, and the deterministic sub-function of $I_2(X)$ which calculates the values V_1 and V_2.

Firstly, the value B_{j_1} is recovered from C_0 by dividing by P and then applying Fermat's Little Theorem using Q. Once B_{j_1} is found it is used as input for $E_1(X)$ and thus the key pair is generated. The ciphertext is XORed, initially with the second secret key, and then with the first. We will assume, as

 ISBN: 978-1-61208-427-5

proposed in Section II.B, that $E_2(X)$ takes as its input a combination of ciphertext length and number of 0's occurring. It is thus easy to determine both of these values, and consequently, to verify the validity of the MACs. If the MACs are valid then the ciphertext may now be "ungarbled" in order to extract the original data; otherwise it is discarded.

Recovering the plaintext is accomplished by using B_{j_1} and the values $C_1, \ldots, C_{N-1}$ given by equations (3) to (5) to obtain $B_1, \ldots, B_{N-1}$. For this procedure it is vital that the sequence of pairs (α_i, j_i) be known. Once each of the values $B_1, \ldots, B_{N-1}$ is obtained, it is simple to remove all inserted bits . Now, the beginning of the ungarbled message is prepended by the first output of $I_2(X)$,so we proceed by removing the V_1-bits. Likewise, the appended V_2-bits is similarly removed. Reversing the circular shift on each block, concatenating the remaining data, and removing the padded zeroes from the final block recovers the original plaintext.

VI. A SIMPLIFIED EXAMPLE

We now give an example of encryption of some sample messages using a method based on a significant simplification of the RGP algorithm. We thank Ritesh Reddy for writing the code which implements this encryption.

```
Number of Blocks: 10     Plaintext Length:
       24     Ciphertext Length: 30
   Original Plaintext: Hello My Name is
                  Ritesh!
  Encrypted:æ└ -°a0\└+¥Êür}næÛ"<<<ñ¬õËËË
    Decrypted: Hello My Name is Ritesh!

 Number of Blocks: 4     Plaintext Length:
       21     Ciphertext Length: 24
Original Plaintext: NSA NSA NSA SECRET!!!
   Encrypted: õúèÇõúcBpucBvhfuhw×××444
     Decrypted: NSA NSA NSA SECRET!!!
```

VII. CONCLUSIONS

In addition to the specific issues noted in previous sections, as an immediate concern, we aim to practically ascertain the relative performance and safety of the RGP versus some of the commonly used cryptographic methods. At the time of writing this paper, this analysis is only preliminary due to lack of a complete working model of the RGP. We also plan to investigate the robustness of the algorithm with regards to accurate decryption when the initial message is comparatively long and to determining how the ciphertext length varies with the number of gates.

References

[1] http://csrc.nist.gov/publications/nist-pubs/800-67-Rev1/SP-800-67-Rev1.pdf (Accessed on July 10, 2015)

[2] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (Accessed on July 10, 2015)

[3] R. C. Merkle, "Secure communications over insecure channels, " Communications of the ACM, Vol. 21, Issue 4, pp. 294-299, 1978.

[4] B. Schneier, J. Kelsey, N. Ferguson, "Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator, " SAC '99 Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography, pp. 13-33, 1999.

[5] G. Hanaoka, J. Shikata, Y. Zheng and H. Imai, “Unconditionally secure digital signature schemes admitting transferability, " ASIACRYPT, LNCS, vol. 1976, pp.130-142, 2000.

[6] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions, " Science, vol. 297, pp. 2026-2030, 2002. [DOI:10.1126/science.1074376]

[7] G. Hanaoka, J. Shikata, Y. Zheng and H. Imai, “Unconditionally secure digital signature schemes admitting transferability, " ASIACRYPT, LNCS, vol. 1976, pp.130-142, 2000.

[8] F. Oggier, H. Fathi, "An authentication code against pollution attacks in network coding, " IEEE/ACM Transactions on Networking, 2009. [DOI:10.1109/TNET.2011.2126592]

[9] A. S. Aiyer, L. Alvisi, R. A. Bazzi and A. Clement, "Matrix signatures: From MACs to digital signatures in distributed systems, " Distributed Computing, 22nd International Symposium, DISC 2008, Arcachon, France, September 22-24, 2008.

 ISBN: 978-1-61208-427-5

A Comparison of the PM-DC-LM Mode With Other Common Operational Block Cipher Modes

Petr Zacek, Roman Jasek, David Malanik
Faculty of Applied Informatics
Tomas Bata University
Zlin, Czech Republic
e-mail: {zacek, jasek, dmalanik}@fai.utb.cz

***Abstract* — The aim of this paper is to compare the performance of Polymorphous Mode - Deterministic Chaos - Logistic Maps (further only PM-DC-LM) with some of the most commonly used block cipher modes of operation. Among the most notable of these are the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). In order to do so, the exclusive OR (XOR) function - instead of a regular block cipher, was used as the encryption algorithm with a Bitmap (BMP) image being used as input data. The testing codes were written in the Python Version 3.4 programming language. Additionally, various operational modes were compared against each other using basic properties like speed or error propagation, among others. The results indicate that PM-DC-LM would seem to be more "random-looking" than the other modes it was tested against and, although the XOR function was used, the output data looked as though it was encrypted. The results also demonstrate the possible advantages in using a polymorphous structure on an image file (i.e. repetitive blocks of data) when compared with the other modes even where the non-standard conditions were set.**

Keywords - cryptography; block cipher; mode of operation; PM-DC-LM; polymorphism; deterministic chaos; logistic map.

I. INTRODUCTION

Most of the generally approved block cipher operational modes are quite old [2]. This is not a disadvantage, but an attempt has been made to do things differently. There are five commonly used operational modes – namely, the ECB mode, the CBC mode, the CFB mode, the OFB mode, and the CTR mode; as described in [2]. All of these block cipher modes provide certain advantages as well as disadvantages, and therefore, their appropriate use or application should always be taken into consideration. For example, the ECB mode was found to be really quick - but not secure, when compared to the other modes; while the CTR mode does show proof of security [8]. These five modes are generally used in order to ensure the confidentiality of encrypted systems. There have been prior attempts at designing new modes [3], however none of them are polymorphous in structure, thereby not offering the chance to explore the PM-DC-LM mode. Moreover, only one mode was designed with the principle of changing the key – i.e. the Key Feedback Mode [7].

Additionally, all of these modes are considered as being fixed to their structure, and are often used as a complement to a block cipher. In view of this, the design of a new group - - called the Polymorphous Mode - Deterministic Chaos - Logistic Map (PM-DC-LM) was explored This mode is derived for example from the Polymorphous Mode (PM); which is not an entirely new proposal. The following section presents a brief introduction of this group (mode) and a brief introduction of some changes that were tested in this paper.

This paper compares the PM-DC-LM mode - as one possible example derived from PM, with the other five previously mentioned modes. The comparison was based on the speed, image data, and other properties of block cipher modes (e.g. parallelizability, error propagation, how a key I affected, security, etc.). This paper – as compared to [1], has been drafted to show the preliminary results of the initial research.

All of these modes were written and tested in the Python, Version 3.4, programming language. The XOR function was used for the "encryption algorithm". It is a well known fact that security is based on the block cipher algorithm. However, attempts were made to test it using the XOR function ("without approved block cipher algorithm").

Section 2 provides a brief description of the PM and PM-DC-LM modes. Section 3 is concerned with the fundamental issues and an introductory comparison. In Section 4, the results arising from testing the PM-DC-LM mode on the base of image data are shown. Section 5 presents the results derived from speed comparisons. In Section 6, the modes are compared on the base of other properties.

II. INTRODUCTION TO (PM) - PM-DC-LM

PM-DC-LM is the acronym for Polymorphous Mode - Deterministic Chaos - Logistic Map. The Polymorphous Mode means that this mode makes variable use of previous plain text, cipher text, and a key to calculate the key for the encryption of the next block of plain text. Deterministic Chaos Mode (DM), on the other hand, is used to determine how the plain text, cipher text and key are used to calculate the key. The Logistic Map represents the type of deterministic chaos.

The PM-DC-LM mode is one possible mode that can be derived from the PM mode. The PM mode represents the main idea about polymorphous modes; therefore, all other modes are fixed to their main structure. This means that they are straightforward and they have only one "way". Our efforts were directed to making a group of modes whose structure is determined by Deterministic Chaos.

 ISBN: 978-1-61208-427-5

This mode can be adapted in many ways because one can change the adjustment of the Chaotic Pseudo-Random Number Generator (CPRNG); or replace Deterministic Chaos with another Pseudo-Random number generator or the function for calculating the new key. This mode is derived from another paper [1], where the mode is described in greater detail. For testing purposes, the chosen modes were almost similar, and adjusted as follows:

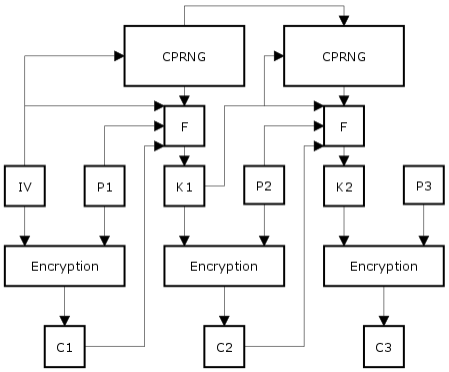


Figure 1. Diagram of PM-DC-LM mode

IV – Initialization vector
C – Cipher text
P – Plain text
K – Key
XOR – Encryption function
F – Function for calculation of the next key
CPRNG – Chaotic Pseudo-Random Number Generator based on a logistic map

A. *Function F*

Function F was slightly changed compared to the original function from [1] and, for testing purposes, the F function was changed as follows:

$$k_{ni} = \begin{cases} (p_i + d + g + i)\ mod\ 256, & for\ d = 1 \\ (c_i + d + g + i)\ mod\ 256, & for\ d = 2 \\ (k_{pi} + d + g + i)\ mod\ 256, & for\ d = 3 \\ (2 \cdot p_i + d + g + i)\ mod\ 256, & for\ d = 4 \\ (2 \cdot c_i + d + g + i)\ mod\ 256, & for\ d = 5 \\ (2 \cdot k_{pi} + d + g + i)\ mod\ 256, & for\ d = 6 \\ (3 \cdot p_i + d + g + i)\ mod\ 256, & for\ d = 7 \\ (3 \cdot c_i + d + g + i)\ mod\ 256, & for\ d = 8 \\ (3 \cdot k_{pi} + d + g + i)\ mod\ 256, & for\ d = 9 \end{cases} \quad (1)$$

k_{ni} – Byte of the next key
k_{pi} – Byte of the previous key
c_i – Byte of the last cipher text
p_i – Byte of the previous plain text
g – The last three digits of value x, generated by CPRNG as a natural number on the interval <1.999>
d – The last digit of value x, generated by CPRNG or g
i – The index of the actual byte

III. A FUNDAMENTAL AND INTRODUCTARY COMPARISON

Fundamental modes manipulate the input for the block cipher, and the block cipher is the main "building block" of these modes. Block ciphers also more or less provide the main security elements. This means that fundamental modes are extremely insecure - without an appropriate block cipher. The PM-DC-LM mode tries to become a different mode - and may provide higher "security levels"; as will be shown in the following section.

Fundamental modes modify the input for the next encryption step. The PM-DC-LM mode tries to manipulate the key for the next encryption step, i.e. the Key FeedBack (KFB) mode – which is described in [7]. Compared to KFB mode, the PM-DC-LM mode enables the use of a variable (polymorphous) structure, which is how the key is computed. Both methods are needed to distinguish cipher texts; even if the plain texts are the same.

The main difference between PM-DC-LM and the other fundamental modes is in its polymorphous or "driven" structure. The PM-DC-LM structure is managed by CPRNG. This means that the structure will have been changed with IV and CPRNG adjustments. Different ways are encapsulated in the F function. As a result, one cannot trace a concrete path without knowledge of the input IV or without knowledge of the CPRNG setting. For all the other modes – i.e. the ECD, CBC, CFB, OFB, and CTR modes, it is possible to trace the way these modes run because it is fixed. The way the PM-DC-LM mode runs varies with the last digit of the CPRNG.

It can be stated that the PM-DC-LM mode is not a "proper" mode like the others, since it incorporates higher functionalities -e.g. CPRNG and the F function. The set goal was to design something new - and different.

IV. COMPARISON OF THE IMAGE DATA BASE

Testing was performed using a BMP image file. The image was composed of 320x320 points in Red/Green/Blue (RGB); which represents 320x320x3 bytes of data, plus 54 bytes for the header. Only image data -> 307200 bytes was used for "encryption purposes" and the XOR function was used as the encryption function - even if this is not standard. The image contained 100x100 points of a black colored square on a white background. The length of the key and blocks was 256 bits (32 bytes). The first key and Initialization vector (IV) were different, and were randomly chosen. The following key in hexadecimal format was used:

91650ae10ea3ca81d629b0c71dc67d063bf215038025d5750 d2c8c5cf7547787

The following IV in hexadecimal format was used:

fce7cf7ffa651ce5d9d56dd92cc49e13bcc3bd17485d75637a8 00f11aea505c8

A. The PM-DC-LM Mode

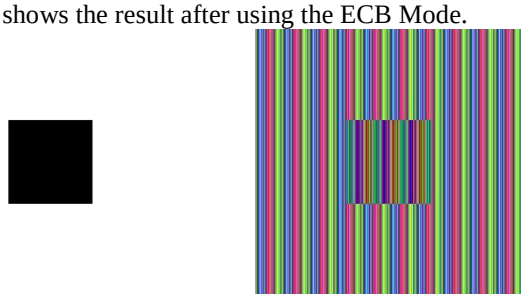
The result, after using the designed PM-DC-LM mode, is shown in Figure 2.

(a) (b)

Figure 2. (a) Original image; (b) Image after using the PM-DC-LM Mode

B. The ECB Mode

Fig. 3 shows the result after using the ECB Mode.

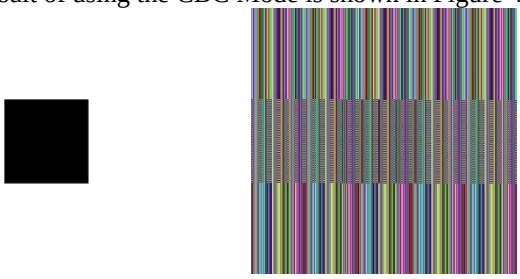


(a) (b)

Figure 3. (a) Original image; (b) Image after using the ECB Mode

C. The CBC mode

The result of using the CBC Mode is shown in Figure 4.

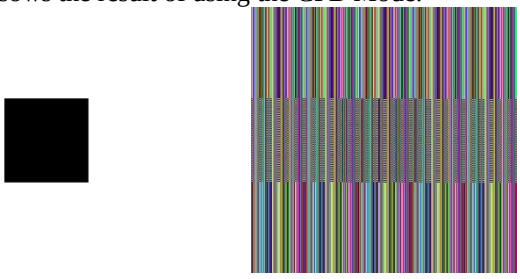


(a) (b)

Figure 4. (a) Original image; (b) Image after using the CBC Mode

D. The CFB Mode shown

Fig. 5 sows the result of using the CFB Mode.

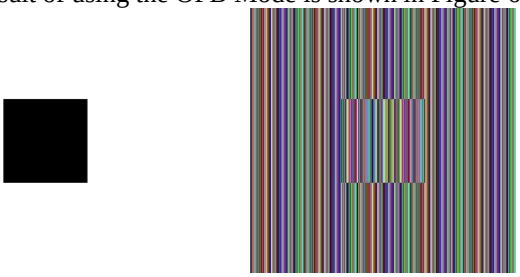


(a) (b)

Figure 5. (a) Original image; (b) Image after using the CFB Mode

E. The OFB Mode

The result of using the OFB Mode is shown in Figure 6.

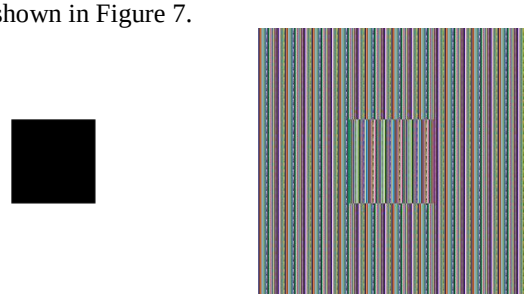


(a) (b)

Figure 6. (a) Original image; (b) Image after using the OFB Mode

F. The CTR Mode

For testing purposes, the CTR Mode was used as follows: the counter was set to number one - represented by a 256-bit number using IV. The result of using the CTR Mode is shown in Figure 7.



(a) (b)

Figure 7. (a) Original image; (b) Image after using the CTR Mode

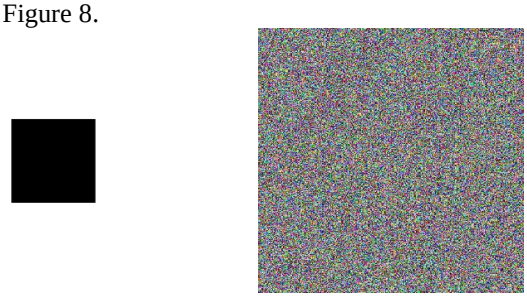
G. Summary and modification

From the images above, one can see that the CBC, CFB, EBC and OFB modes look more or less alike; while the CTR Mode is a little different; where the difference is in the last bytes of the cipher text blocks. This is because of the counter. Only, the PM-DC-LM Mode - without using a block cipher, looks like “random noise” - excluding blocks with similar shades. These blocks are due to the use of plain text (all zeros or ones in the BMP image) for the calculation of the next key for the encryption phase. As a result, the following key will be calculated using the same bytes. This deficit could be ameliorated by the modification of the first, fourth and seventh equations in the F function by the addition of the byte of the previous cipher text. The F function can be modified as follows:

$$k_{ni} = \begin{cases} (p_i + c_i + d + g + i) \bmod 256, & for\ d = 1 \\ (c_i + d + g + i) \bmod 256, & for\ d = 2 \\ (k_{pi} + d + g + i) \bmod 256, & for\ d = 3 \\ (2 \cdot p_i - c_i + d + g + i) \bmod 256, & for\ d = 4 \\ (2 \cdot c_i + d + g + i) \bmod 256, & for\ d = 5 \\ (2 \cdot k_{pi} + d + g + i) \bmod 256, & for\ d = 6 \\ (3 \cdot p_i + 2 \cdot c_i + d + g + i) \bmod 256, & for\ d = 7 \\ (3 \cdot c_i + d + g + i) \bmod 256, & for\ d = 8 \\ (3 \cdot k_{pi} + d + g + i) \bmod 256, & for\ d = 9 \end{cases} \quad (2)$$

 ISBN: 978-1-61208-427-5

The result of using the modified PM-DC-LM Mode is shown in Figure 8.



(a) (b)

Figure 8. (a) Original image; (b) Image after using the modified PM-DC-LM Mode

This result looks better when compared to the result in Figure 1, since the function was modified by adding cipher text which the F function computes with plain text (for d = 1, d = 4, and d = 7)

V. SPEED COMPARISON

This section analyses and compares the time for "encryption" - dependent on the mode. The time depends on the implementation of the mode; but for testing purposes, all modes were implemented in a similar manner. All modes were implemented in Python, Version 3.4 and were written without using parallelizability.

TABLE I. TIME NEEDED FOR "ENCRYPTION" BY MODE

Time for "encryption"			
	BMP 320x320	1 MB Image	10 MB Image
PM-DC-LM	1.163 s	3.786 s	37.858 s
ECB*	0.100 s	0.326 s	3.255 s
CBC	0.182 s	0.592 s	5.924 s
CFB	0.191 s	0.622 s	6.217 s
OFB	0.184 s	0.599 s	5.990 s
CTR*	0.288 s	0.938 s	9.375 s

* Without parallelization

In Table 1, we can see that the PM-DC-LM mode is approximately four times slower than the CTR mode and from six to seven times slower than the CBC, CFB, and OFB modes, and eleven times slower than the ECB mode.

VI. COMPARISON ON THE OTHER PROPERTIES

A. *"Security Level"*

It is a well known fact that the security of an encryption algorithm is mainly based on the block cipher. Thus, the comparison based on this property could be misleading; despite this, one can try to compare it. For testing purposes, the security level will be measured against these conditions:

- Chaining (i.e. the previous block affects the encryption of the next block) - Positive
- IV (i.e. the insertion of another random factor) - Positive
- The level of dependency on block cipher security - Negative
- Extension (i.e. Deterministic Chaos) – Positive

From the conditions above, one could derive that the ECB mode should be the least secure; followed by the CTR mode. The OFB, CBC, and CFB modes follow; all have similar security levels. The PM-DC-LM mode should have the highest security level.

In the PM-DC-LM mode, security depends on the CPRNG and, according to [4], logistic maps may be used as a CPRNG. But this could be changed in an appropriate manner or another type of Deterministic Chaos could be used.

Another important thing is that the IV must be random in order to achieve "indistinguishability" from random bits, used only once.

According to [6], the PM-DC-LM mode has no CCA security; and the mode is secure as a probabilistic encryption scheme. The security level will vary according to the different design of the CPRNG.

No attack using a knowledge of the structure may be used, since the structure cannot be known without a knowledge of the IV or of the CPRNG adjustement. This may be the greatest contribution of this mode.

B. *Errors*

The errors depend upon the place and time of the error occurrence. Only bit errors (where the bit is changed from 0 to 1 or from 1 to 0) during transmission after encryption can be considered. Two sample cases are especially discussed here, (an error in IV, or error in a block of cipher text).

1) *Error bit in IV*

The ECB mode does not have IV. Thus, ECB caused by errors in IV cannot occur. For the CFB and CBC modes, the first block will be decrypted incorrectly whilst in the CTR (where IV is used) and the OFB, and PM-DC-LM modes, all blocks will be decrypted incorrectly.

2) *Error bit in a block of cipher text*

For the ECB, CTR, and OFB modes, the error in a block of cipher text does not affect the decryption of the other blocks. The error will only be in the block corresponding to the block of cipher text with an error bit.

For the CFB and CBC modes, the error in a block cipher affects all of the other decrypted blocks - including any block corresponding to a block of cipher text with an error bit.

For the PM-DC-LM mode, there is a 33.3 % probability that the error bit in the block of the cipher text will not affect the other blocks during decryption.

C. *Parallelizability*

The ECB and CTR modes can be fully parallelized. The CBC and CFB modes can only be parallelized during

decryption. The OFB mode cannot be parallelized at all. In PM- DC-LM mode, the CPRNG and encryption can be separately computed and cannot be parallelized.

D. *Affecting the key*

All of the modes compared above - excluding the PM-DC-LM mode, do not affect the key(s) for encryption.

E. *Summary*

All the other properties are summarized in Table 2.

TABLE II. SUMMARIZATION OF THE COMPARISON OF THE OTHER PROPERTIES

Mode	Other properties			
	Parallelizable		Error propagation	Chaining
	Encryption	Decryption		
PM-DC-LM	No	No	Yes	Yes
ECB	Yes	Yes	No	No
CBC	No	Yes	Yes	Yes
CFB	No	Yes	Yes	Yes
OFB	No	No	Yes	Yes
CTR	Yes	Yes	No	No
Mode	Affecting of key	IV/Nonce	The level of dependency on the block cipher security	The level of security without block cipher
PM-DC-LM	Yes	Yes	Low	High
ECB	No	No	High	Very low
CBC	No	Yes	High	Very low
CFB	No	Yes	High	Very low
OFB	No	Yes	High	Very low
CTR	No	Yes	High	Very low

VII. CONCLUSION

In this paper, the authors have tried to compare their own design of a block cipher mode of operation - called PM-DC-LM as an example derived from PM with other modes. Specifically, comparisons were made based on the speed and image data using the XOR function as the encryption algorithm instead of a regular block cipher, even if it is "extra-ordinary". Using the Advanced Encryption Standard (AES), the result would be different. Additionally, basic properties like parallelizability, "security" level and propagation errors were explored.

The results of these comparisons indicate that the PM-DC-LM mode may be more secure if one uses random IV and an appropriate design for the CPRNG or other PRNG. The authors´ also realized that the PM-DC-LM mode is slower - compared with the other modes, and behaves "randomly". The PM-DC-LM mode is more prone to error propagation and cannot be parallelized. By changing the F function, the behavior of the mode was changed as was illustrated on the image data.

The PM-DC-LM mode would appear to be a potent mode for an "encryption" algorithm; but the results may be better using block ciphers instead of the XOR function. This mode may be immune to all attacks based on structure.

Since this is a preliminary work, only basic comparisons were made. Future work hopes to continue the research with some interesting findings.

ACKNOWLEDGMENT

This work was supported by the Tomas Bata University Internal Grant Agency, Project No.: IGA/FAI/2015/47; further, it was supported by financial support from the Ministry of Education of the Czech Republic research project NPU I No.: MSMT-7778/2014; as well as by the European Regional Development Fund, CEBIA-Tech Project No.: CZ.1.05/2.1.00/03.0089

REFERENCES

[1] P Zacek, R. Jasek, and D. Malanik, "Using the deterministic chaos in variable mode of operation of block ciphers", in Artificial Intelligence Perspectives and Applications (CSOC 2015), Springer International Publishing, 2015 pp. 347-354, doi:10.1007/978-3-319-18476-0_34.

[2] Current Modes. In: Special Publication 800-38A: First Part: Five Confidentiality Modes 2001. [Online]. Available from: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[3] Modes Development. In: National Institute of Standards and Technology: Computer Security Resource Center [online]. 2001, http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html

[4] J. C. Sprott, Chaos and Time-Series Analysis, Oxford University Press, 2003

[5] R. Senkerik, M. Pluhacek, I. Zelinka, D. Davendra, and Z. Oplatkova, "A brief survey on the chaotic systems as the pseudo random number generators",, in Interdisciplinary Symposium on Complex Systems, vol 14. Emergence, Complexity and Computation (ISCS 2014),, Springer International Publishing, 2015, pp. 205-214, doi:10.1007/978-3-319-10759-2_22

[6] P. Rogaway, "Evaluation of some block cipher modes of operation", Feb. 2011. [Online]. Available from: http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf

[7] J. Hastad and M. Näslund, Key Feedback Mode: a Keystream Generator with Provable Security, Oct. 2000. [Online]. Available from: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/kfb/kfb-spec.pdf

[8] H. Lipmaa, P. Rogaway, and D. Wagner. "Comments to NIST concerning AES-modes of operation: CTR-mode encryption", in Symmetric Key Block Cipher Modes of Operation Workshop, Baltimore, Maryland, US, 2000. [Online]. Available from: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/workshop1/papers/lipmaa-ctr.pdf

 ISBN: 978-1-61208-427-5

An Improved Threshold Proxy Signature Scheme

Akanksha Gupta, Prakash D.Vyavahare, Manish Panchal

Department of Electronics and Telecommunication Engineering
S. G. S. Institute of Technology and Science, Indore, India

Email: akkuregister.90@gmail.com, prakash.vyavahare@gmail.com, hellopanchal@gmail.com

***Abstract*—Threshold proxy signature schemes allow original signer to delegate his signing capability to a group of *n* proxy signers in which *t*, $(1 < t \leq n)$ or more proxy signers out of *n* can generate a valid proxy signature on behalf of original signer. The first Rivest, Shamir and Adleman (RSA) based threshold proxy signature scheme was proposed by Hwang et. al. . Later on Wang in his paper, commented on the security weakness in Hwang's scheme. In this paper, we propose a new RSA based scheme that removes weakness of Hwang's scheme. The proposed scheme employs efficient generation of shared RSA key using algorithm proposed by Boneh and Fanklin and RSA threshold cryptosystem of Nguyen. The proposed scheme provides secrecy, unforgeabilty, nonrepudiation, proxy protection and removes the need of trusted combiner. Therefore, it can be a potential candidate for implementation of electronic proxy signature system.**

Keywords– Proxy Signature; Threshold Signature; Secret Sharing; RSA.

I. INTRODUCTION

Digital signature is a cryptographic scheme used to authenticate the identity of the sender of a message and sender can not repudiate the message once it is signed by him. It also assures the recipient for the integrity of message. Many practical applications implement it either directly or in some other form. Proxy signature is one such example, where digital signature on a message is performed by *proxy signer* on behalf of original signer in his absence, to whom original signer has delegated his signing power.

Mambo *et. al.* [1] in 1996 first proposed the concept of proxy signature and classified it on the basis of delegations namely, *full delegation*, *partial delegation* and *delegation by warrant*. Full delegation was not secure and hence was not used in practical systems. In such a case, original signer gives his private key to proxy signer. Therefore, proxy signature is indistinguishable from original signature. Partial delegation and delegation by warrant are more secure than full delegation scheme.

In 1997, Kim [2] combined the idea of secret sharing and threshold crypto system to proxy signature scheme with less trust on single proxy signer. Initially proposed threshold proxy signature schemes were based on principle of discrete logarithm cryptosystem [3][4]. In (t, n) threshold scheme, t number of proxy signers $(1 < t \leq n)$ can cooperatively generate valid proxy signature on a message, but $(t-1)$ or less proxy signers can not generate valid sign on the document. Threshold proxy signature scheme is more secure and practical than conventional proxy signature scheme. Number of publications of threshold proxy signature based on discrete logarithm have been reported [5][6][7][8]. Hwang et. al. [9] proposed the first RSA based threshold proxy signature scheme in which author described six requirements that should be satisfied by a secure (t, n) threshold proxy signature scheme.

1) **Secrecy:** Original signer's private key must be kept secret with original signer and it should not be possible to derive it by any proxy signer, not even by cooperation among them.
2) **Proxy protected:** Partial proxy signature of a designated proxy signer can only be generated by him. Even original signer can not be masquerade partial proxy signature. Partial proxy signature key must not known to original signer.
3) **Unforgeability:** Only t or more designated proxy signers can cooperatively generate valid proxy signature.
4) **Non repudiation:** Once t or more proxy signers cooperatively generate valid proxy signature, they can not deny their signatures and original signer also can not deny delegating the signing power to the proxy signers.
5) **Time constraint:** The proxy signature key can be used during the delegation period. After this period, proxy signature generated by proxy signer will be considered to be invalid.
6) **Known signers:** Scheme must be able to identify the actual group of signers from proxy group in threshold scheme.

Wang [10] analyzed the security aspects of Hwang's scheme and commented that Hwang's scheme was unable to fulfill the security requirements of threshold proxy signature system. Number of publications of the threshold proxy signature scheme based on RSA has been reported [11][12][13][14], that modified the Hwang's scheme and removed various security weakness. The most recent publication [15] adds a new feature by allowing n proxy signers, renew their own proxy shares periodically without changing the secret. In the proposed scheme the combiner need not be trusted. The verifier can independently verify if the combiner has done untrustworthy operation on proxy signature. All the above proposed schemes require trusted combiner and trusted original signer for generation and verification secret shares.

In this paper, we propose an improved threshold scheme based on RSA algorithm [16]. In the proposed scheme, n proxy signers execute the Boneh and Franklin protocol [17] to generate RSA based signature scheme modulo N, whose prime factors are not known to proxy signers and they do not

 ISBN: 978-1-61208-427-5

TABLE I. CONVENTION AND NOTATION

C	Combiner
D	Proxy signature key to generate proxy signatures
D_i	Additive share of D for S_i
E	Proxy verification key to verify proxy signatures
N	RSA modulo
S	Proxy signature on message m
R_i	Random number
S_0	Original signer
S_i	i^{th} Proxy signer $(1 \leq i \leq n)$
T	Subset of n Proxy Signers who cooperatively generate signature on message m
V	Verifier
d_0	Original signer's private key
d_i	i^{th} Proxy signer's private key
e_0	Original signer's public key
e_i	i^{th} Proxy signer's public key
k_i	Partial proxy signature key of S_i to generate partial proxy signature
m	message to be signed
n	Number of proxy signer
r_i	Random number
s_i	Partial signature of S_i on message m
t	Threshold number, $(1 < t \leq n)$ of proxy signers require to generate proxy signature on message m
w	warrant generated by S_0
$\phi(N)$	Euler totient function
$\|$	concatenation of bit strings

know $\phi(N)$. Each proxy signer will derive its additive share D_i for Proxy signature key D without revealing it to others, such that $D = D_1 + D_2 + ... + D_n$. These shares are used to compute partial proxy signature, which are used to generate proxy signature S later on the need of trusted combiner has been removed in the proposed scheme.

The rest of the paper is organized as follows. Hwang's scheme is reviewed in Section II. Section III describes the weakness of Hwang's scheme. In Section IV proposed improved threshold proxy signature scheme is presented which is analyzed in Section V. Finally, the paper is concluded in section VI.

II. REVIEW OF HWANG'S SCHEME

Let S_0 denote the original signer and $S_1, S_2.....S_n$ be *n* proxy signers. C is a signature combiner and V is a signature verifier. Let $N_i = p_i * q_i$ be a public RSA modulus for S_i, where p_i and q_i are two secret large prime numbers for i= 1, 2, 3,..., n. Each of the proxy signers S_i has its own public key e_i and private key d_i such that $d_i * e_i = 1 mod \phi(N_i)$. Let d_0 and e_0 be the private and public key of original signer and w is be a warrant generated by S_0. The warrant includes delegation period for validity of proxy signature, the proxy signers' identities and identity of the original signer. Other notations and conventions are shown in Table I.
Hwang's scheme consists of three phases which are shown in Figure 1 :

A. Proxy sharing phase: The original signer generates share of proxy signature key D.
B. Proxy signature issue phase: Proxies generate partial proxy signatures which, after combining, will create proxy signature on message and
C. Verification phase: Proxy signature is verified by proxy verification key E.

A. The Proxy Sharing Phase

The original signer S_0 delegates his signing capability to *n* proxy signers as follows:

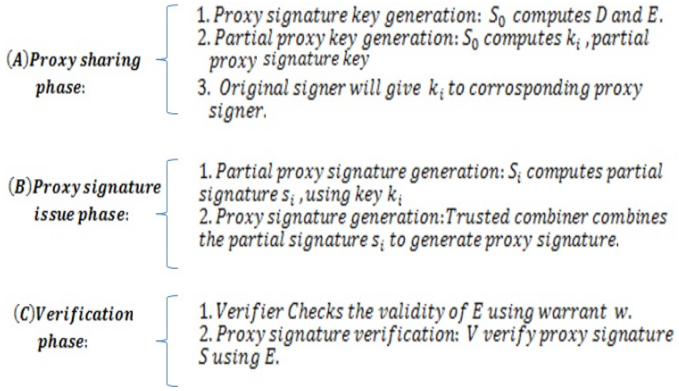


Figure 1. Three phase of Hwang's Scheme

1) S_0 computes $D = d_0^w mod \phi(N_0)$ and $E = e_0^w mod \phi(N_0)$ as proxy signature key and proxy verification key respectively. S_0 publishes w, E, and (w $\|$ E)$^{d_0} mod N_0$.

2) S_0 generates partial proxy signature key k_i, shares of secret D for corresponding proxy signer S_i, using Shamir's secret sharing scheme [18]. For this, S_0 selects a polynomial

$$f(x) = D + R_1 x + ... + R_{t-1} x^{t-1} mod \phi(N_0) \quad (1)$$

of degree $(t-1)$, where $R_1, R_2, ...R_{t-1}$ are random numbers chosen by S_0. Then, S_0 computes

$$k_i = f(i) \quad (2)$$

for S_i, where $i = 1, 2, ...n$ and sends $(k_i{}^{d_0} mod N_0 \| k_i)^{e_i} mod N_i$ to S_i on public channel.

3) After receiving the data from S_0, S_i computes k_i by first decrypting it with its private key d_i and then by public key e_0 of S_0 as $k_i = (((k_i{}^{d_0} mod N_0 \| k_i)^{e_i} mod N_i)^{d_i} mod N_i)^{e_0} mod N_0$, where $i = 1, 2, ...n$.

B. The Proxy Signature Issue Phase

To sign a the message m on behalf of original signer S_0, any t or more proxy signers form a subset T of proxy signers. They generate proxy signature as follows:

1) Each proxy signer of subset T computes its partial proxy signature s_i on message m as follows:

$$s_i = m^{L_i k_i} mod N_0 \quad (3)$$

where, i indicates the i^{th} proxy signer from subset T. L_i is Lagrange interpolation coefficient given by

$$L_i = \prod_{i,j \epsilon T, j \neq i} \frac{-j}{i-j} \quad (4)$$

and sends $s_i \| s_i{}^{d_i} mod N_i$ to combiner C.

2) Combiner verifies s_i using public key e_i of S_i and then computes proxy signature S as follows:

$$S = \prod_{i=1}^{n} s_i mod N_0 = m^D mod N_0 \quad (5)$$

C. The Verification Phase

After receiving proxy signature S, the verifier V verifies it as follows:

1) V receives w, E, (w $\|$ E)$^{d_0} mod N_0$ from S_0 and verifies it using $(w\|E) = ((w\|E)^{d_0} mod N_0)^{e_0} mod N_0$. V first checks the validity of E by checking the valid period mentioned in the warrant. If the period has expired then E is invalid and it can not be used for verification.

2) If E is valid, then V computes $S^E mod N_0$ and checks whether it is equal to m.

$$S^E mod N_0 = m \quad (6)$$

Since
$S^E mod N_0 = (m^D)^E mod N_0$
$= m^{d_0{}^w e_0{}^w} mod N_0$
$= m^{(d_0 e_0)^w} mod N_0 = m$, and

3) The actual proxy signers identity can be identified using his public key e_i on ${s_i}^{d_i} mod N_i$.

III. SECURITY ANALYSIS BY WANG

Wang [10] claimed that Hwang's scheme was not able to satisfy the security requirements of threshold proxy signature scheme and indicated various security weaknesses in it, such as:

- Secrecy: Proxy signers from the set T can cooperatively compute proxy signature key D and $(DE - 1)$, factors of $\phi(N_0)$. Knowing these factors is equivalent to factoring N_0 [21]. Finally, with the factors of N_0, the proxy signers can compute the value of $\phi(N_0)$. Once factors of $\phi(N_0)$ are calculated, it is easy to calculate d_0 by using $d_0 e_0 = 1 mod \phi(N_0)$. Hence, private key of original signer will not remain secret.

- Proxy Protected: Original signer knows the partial signing key k_i of corresponding proxy signer S_i. Therefore, he can create the partial proxy signature s_i on message m on behalf of S_i as k_i is just the share of D created by S_0 himself. Another security weakness is that the proxy signers have to trust on original signer that given partial proxy signature key k_i is correct for generating valid proxy signature S.

- Unforgeability: Hwang's scheme is weak in yet another aspect that an unauthorized third person can compute proxy signature S as he can calculate factors of N_0 because $({e_0}^w - E)$ is also a factor of $\phi(N_0)$. Using similar strategy as mentioned above, he can compute d_0 as well as $D = d_0^w mod \phi(N_0)$. However, it is a time consuming process.

- Non Repudiation: D is generated by original signer. Therefore, he can create proxy signature S on message m as $m^D mod N_0$ by surpassing the combiner. Moreover, t proxy signers can also cooperatively compute proxy signature without combiner. In such cases Verifier will accept a signature S by checking $(S^E mod N_0 = m)$. In future, if such a valid proxy signature S causes dispute on the identity of signer of message, it can not be found out as who was responsible for generating signature, whether the original signer or proxy signers.

- Known Signer: In Hwang's scheme, a trusted combiner is required. Otherwise, partial proxy signature s_i can be altered, replaced or deleted by the combiner.

IV. PROPOSED SCHEME

The proposed scheme consists of four phases namely (A). Initialization phase, (B). Proxy sharing phase, (C). Proxy signature issue phase, (D). Verification phase. These phases are described as follows in Figure 2:

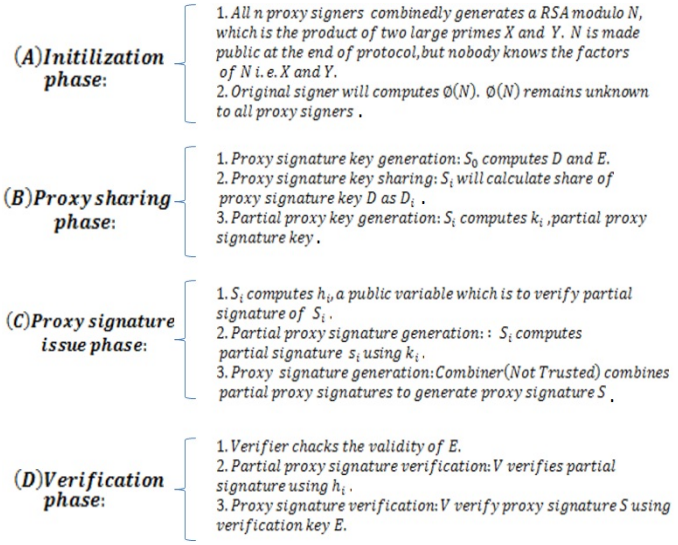


Figure 2. Four phases of the proposed scheme

A. Initialization Phase

1) All n proxy signers combinedly generate a number N, which is the product of two large primes by employing Boneh and Franklin protocol [17]. N is made public but nobody knows the factors of N at the end of protocol. Following steps are followed to generate N :
Participants secret and Distributed Sieving:
   a) Each proxy signer S_i randomly picks up two secret numbers x_i, y_i.
   b) All proxy signers determine whether or not the sums $X = \sum_{i=1}^{n} x_i$, $Y = \sum_{i=1}^{n} y_i$ are not divisible by any prime number between 0 and some bound B1.

Computation of N: All proxy signers will compute N without reveling any information about their secrets x_i and y_i . They agree on a big prime number $M > N$ and an element g of high order in Z_N^* and use BGW protocol [19] as follows to generate N.

$$N = \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i = X * Y mod M \quad (7)$$

Trial Division: This is required to ensure that N is not divisible by any number between B1 and B2, numbers which are agreed by all the parties.

Primality Test: Extended Fermat's Primality test is performed to check whether N is a product of two prime numbers or not. If it is not, then protocol is repeated from the first stage with new values of x_i, y_i until it passes the primality test.

2) Computation of $\phi(N)$

a) Each proxy signer S_i will calculate ϕ_i, the share of $\phi(N)$ as follows:

$$\phi_i = \{^{N-x_i-y_i+1 \quad if \ i=1}_{-x_i-y_i \quad if \ i>1} \quad (8)$$

and sends it to S_0 as $(\phi_i \| \phi_i^{d_i} mod N_i)^{e_0} mod N_0$.

b) After receiving data from all S_i, the original signer extracts ϕ_i by decrypting it with the private key d_0 of original signer and public key e_i of corresponding i^{th} proxy signer. S_0 then computes $\phi(N)$ as:

$$\phi(N) = \sum_{i=1}^{n} \phi_i$$
$$= N - \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i + 1$$
$$= (X-1)(Y-1)$$

Note that proxy signers can not determine $\phi(N)$ unless all proxy signers combine their shares ϕ_i for the generation of $\phi(N)$. Therefore $\phi(N)$ is (n-1) private.

B. Proxy Sharing Phase

This phase involves generation of proxy signature key D and its corresponding verification key E, by S_0 and generation of partial proxy signature key k_i by S_i. Note that k_i is generated by S_0 in Hwang's scheme.

Proxy Signature Key Generation:

1) S_0 chooses a random number a, where $a \epsilon Z_M$ and $gcd(a, \phi(N)) = 1$, and calculates $b = a^{-1} mod \phi(N)$ using Euclids extended algorithm.
2) Original signer then computes proxy signature key D and proxy verification key E, as follows:

$$D = b^w mod \phi(N), E = a^w mod \phi(N) \quad (9)$$

$DE = 1 mod \phi(N)$
3) S_0 will compute $\psi = \phi(N) mod E$ and $\psi^{-1} mod E$.
4) Proxy signature key D remains secret with original signer, Corresponding verification key E is made public for the verification of proxy signature.
5) S_0 publishes $w, E, a, \psi, (w \| E \| a \| \psi)^{d_0} mod N_0$,

Proxy Signature Key Sharing:

1) Each Proxy signer S_i receives the published $w, E, a, \psi, (w \| E \| a \| \psi)^{d_0} mod N_0$, from original signer, and verifies that $(w \| E \| a \| \psi) = ((w \| E \| a \| \psi)^{d_0} mod N_0)^{e_0} mod N_0$ using public key e_0 of original signer.
2) S_i gets proxy verification key E and warrant w, Therefore, each proxy signer can check the validity of E by checking the valid period mentioned in warrant w. If E finds it to be valid, then proxy signer will accepts it to derive shares of proxy signature D. Otherwise, he will reject it and request S_0 for a valid warrant and signature. Else, he stops this protocol.
3) Each S_i will calculate the additive share D_i of proxy signature key D using Boneh and Franklin protocol [17].

Each S_i will compute D_i as follows:

$$D_i = \{^{(1-\phi_1 \psi^{-1})/E \quad if \ i=1}_{-(\phi_i \psi^{-1})/E \quad if \ i>1} \quad (10)$$

Finally, D is calculated by S_0 as follows:

$$D = \sum_{i=1}^{n} D_i =$$
$$(1 - \phi_1 \psi^{-1})/E + \sum_{i=2}^{n} (-\phi_i \psi^{-1})/E.$$

Calculated value of D is kept secret with S_0.

Partial Proxy Signature Key Generation: Proxy signers compute their own partial proxy signature keys k_i, as the share of actual proxy signature key D. However, D is not known to the proxy signers. k_i is calculated by S_i using following steps:

1) Each S_i selects a random polynomial $f_i(x) \epsilon Z_M$ of degree $(t-1)$, with $f_i(0) = D_i$. Let $f_i(x)$ be

$$f_i(x) = D_i + r_{i,1} x + ... + r_{i,t-1} x^{t-1} \quad (11)$$

S_i proxy signer computes $f_{i,j} = f_i(j)$ which is the share of D_i for S_j and sends $(f_{i,j} \parallel {f_{i,j}}^{d_i} mod N_i)^{e_j} mod N_j$ to S_j for $1 \leq j \leq n$ on the public channel. S_i broadcasts $c_{i,j} = g^{r_{i,j}} mod N$ for j = 0, 1...., (t-1) to others.
2) S_j verifies validity of the share $f_{i,j}$ received from S_i, using following formula :

$$g^{f_{i,j}} = g^{f_i(j)} = g^{D_i + r_{i,1} j + ... + r_{i,t-1} j^{t-1}} mod N$$
$$= g^{D_i} . g^{r_{i,1} j} ..... g^{r_{i,t-1} j^{t-1}} mod N$$
$$= \prod_{k=0}^{t-1} c_{i,k}^{j^k} mod N$$

If verification fails, then S_j sends error message to the original signer.
3) Partial proxy signature key k_i for S_i is calculated as

$$k_i = \sum_{j=1}^{n} f_j(i) \quad (12)$$

k_i is kept as secret by S_i.
4) The above computation of k_i, as done by corresponding proxy signer S_i, is same as the one obtained by original signer S_0 in Hwang's scheme from polynomial $f(x) = D + R_1 x + ... + R_{t-1} x^{t-1}$ of degree (t-1) having $f(0) = D$ and $k_i = f(i)$. The group of n proxy signers can obtain a polynomial f(x) of at most $(t-1)$ degree in the following form :

 ISBN: 978-1-61208-427-5

$f(x)$
$= \sum_{j=1}^{n} D_i + (\sum_{j=1}^{n} r_{j,1})x + ...$
$.. + (\sum_{j=1}^{n} r_{j,t-1})x^{t-1}$
$= D + R_1 x + ... + R_{t-1}x^{t-1}$
$k_i = f(i)$

C. Proxy Signature Issue Phase

1) Each proxy signer S_i computes

$$\sigma_i = L_i * k_i mod M \tag{13}$$

for $i = 1, 2, ..., n$. L_i is Lagrange interpolation coefficient which is calculated as follows:

$$L_i = \prod_{i,j \epsilon T, j \neq i} \frac{-j}{i-j} mod M \tag{14}$$

σ_i is kept secret by corresponding proxy signer S_i.

2) To sign the message m, proxy signers S_i, from the set T, will sign on behalf of original signer by generating his partial proxy signatures s_i on m as follows:

$$s_i = m^{\sigma_i} mod N \tag{15}$$

where $i \epsilon T$

3) Each proxy signer computes $h_i = g^{\sigma_i} mod N$, which will be used to verify the signature shares generated by proxy signers at verification stage.
4) S_i publishes its partial signature s_i, h_i and $(s_i \| h_i)^{d_i} mod N_i$.
5) Now a combiner (not necessary to be a trusted one) or any one proxy signer from group of n proxy signers, will combine all partial signatures to generate proxy signature S by collecting s_i's, and calculates as follows:

$$S = \prod_{i \epsilon T} s_i mod N \tag{16}$$

$= \prod_{i \epsilon T} m^{L_i k_i} mod N = m^{\sum_{i \epsilon T} D_i} mod N = m^D mod N$ The combiner publishes message m with its signature S and $(s_i \| h_i)^{d_i} mod N_i (i \epsilon T)$ to the verifier.

D. Verification Phase

To verify the signature S on message m, the verifier V will carry out the following steps:

1) The verifier V receives $w, E, a, \psi, (w \| E \| a \| \psi)^{d_0} mod N_0$, from S_0 and verifies that $(w \| E \| a \| \psi) = ((w \| E \| a \| \psi)^{d_0} mod N_0)^{e_0} mod N_0$. Verifier will also check the validity of E by checking the valid period mentioned in the warrant. If the period has expired, E is invalid and can not be used for verification.
2) The verifier V also receives $(s_i \| h_i)^{d_i} mod N_i (i \epsilon T)$. He retrieves s_i by decrypting with the public key e_i of S_i and verifies partial signatures as

$$Dlog_m s_i = \sigma_i \tag{17}$$

$= Dlog_g (g^{s_i})$
$= Dlog_g (h_i)$

3) After verifying partial signature, V will verify proxy Signature S using E. Finally, the message m is retrieved as

$$S^E mod N = m^{DE} mod N = m \tag{18}$$

V. SECURITY ANALYSIS

In this section, it is shown that the proposed scheme satisfies all security requirements of threshold proxy signature scheme.

1) **Secrecy:** In Hwang's scheme, proxy signature key D is computed with the original signer's private key d_0, as
$$D = d_0^w mod \phi(N_0)$$
which can be cooperatively computed by proxy signers. Wang explained that knowing D and $(DE - 1)$, it is easy to calculate d_0 with $d_0 e_0 = 1 mod \phi(N_0)$.
In our scheme, proxy signature key D is computed with a new randomly selected number b by S_0 as follows.
$$D = b^w mod \phi(N)$$
Original signer's private key d_0 remains secret with original signer and it is not shared with others in any phase in the proposed scheme.
2) **Proxy Protected:** Proxy protection requires that only a designated proxy signer S_i can generate his partial proxy signature s_i. However, in Hwang's scheme it was found that partial proxy signature key k_i was just a share of D generated by the original signer S_0 and it not derived from private key d_i of proxy signer S_i. Hence, k_i is known to original signer. Therefore, he can compute s_i which is partial signature of i^{th} proxy signer. In Hwang's scheme, original signer needs to be trusted one for sending signing key to proxy signer S_i. The proposed scheme does not require a share of D from trusted original signer.
In the proposed scheme, no one else can generate valid partial proxy signature s_i because it is computed using k_i as $s_i = m^{L_i * k_i} mod N$, which remains secret with respective proxy signer S_i. In our scheme partial proxy signature key k_i is derived by i^{th} proxy signer and k_i remains unknown to original signer. The proposed scheme require $n(n-1) + n$ additional transmission for generation of k_i. In Hwang's scheme, n transmission are required for generation of k_i. Hence, original signer can not forge the partial proxy signature of $S_i$in our scheme.
3) **Unforgeability:** A valid proxy signature is generated by the cooperation of t or more proxy signers. A non designated third party can not forge proxy signature S as he can not derive D. In the Hwang's scheme, original signer S_0 (who knows D), can forge a signature on message m as $S = m^D mod N$. Verifier will accept a proxy signature S just by checking whether it satisfies the equation $S^E mod N = m^{DE} mod N = m$ or not.
In the proposed scheme, the verifier V needs s_i for verification of the partial proxy signature of the corresponding proxy signer at verification stage.

 ISBN: 978-1-61208-427-5

Hence, scheme collects all partial proxy signature s_i at verification stage from a combiner. The combiner need not be trusted party. The verifier at later stage can find out if the combiner has modified or delete any partial signature. Original signer can not compute partial signature s_i because he does not know k_i to calculate $s_i = m^{k_i * L_i}$, which is required at verification stage.

4) **Non Repudiation:** Each proxy's partial signature s_i is encrypted by its private key d_i which is computed by proxy signer S_i. Hence he can not deny the existence of proxy signature on signed message. Original signer can not deny delegating the signing right as he publishes warrant w in the form $(w\|E\|a\|\psi)^{d_0} mod N_0$ which is encrypted using its private key d_0.
5) **Known Signer:** For internal auditing, combiner need not be trusted in the proposed scheme. It combines partial signatures of proxy signers. If combiner tries to modify or delete s_i, it will be detected by verifer at verification phase, as it is encrypted with the proxy's private key.
6) **Time Constraint:** Original signer publishes warrant w with proxy verification key E in the form $(w\|E\|a\|\psi)^{d_0} mod N_0$. Verifer will check the validity of E in the delegation period provided in warrant w to check whether E is the valid proxy verification key.

VI. PERFORMANCE COMPARISON

The performance comparison of the proposed scheme with the Hwang's scheme is shown in Table II. It can be noted that proposed scheme has merits of proxy protection since only a designated proxy signer S_i can generate his partial proxy signature s_i; non designated third party can not forge proxy signature This also achievers non-repudiation. In addition, the proposed scheme does not require trusted original signer and trusted combiner. However, the proposed scheme requires additional initialization phase. Thus, it can be seen that the proposed scheme has numerous cryptoanalytical merits as compared to that of Hwang's method.

TABLE II. COMPARISION OF PROPOSED SCHEME WITH HWANG'S SCHEME

S.No	Security Requirement	Hwang's Scheme	Our Scheme
1	Secrecy	No	Yes
2	Proxy Protected	No	Yes
3	Unforgeabilty	No	Yes
4	Non Repudiation	No	Yes
5	Time Constraint	Yes	Yes
6	Known Signer	Yes	Yes
7	Trusted Combiner	Yes	No
8	Secure Channel	No	No

VII. CONCLUSION

In this paper, we have proposed a new threshold proxy signature scheme based on RSA, which uses the Boneh and Franklin protocol and Hguyens scheme for generation of shared RSA keys. It has been shown that the proposed scheme significantly removes the weaknesses found by Wang in the Hwang's scheme which is also based on RSA. The marginal increase in computational complexity in the proposed scheme is compensated by reduction in the cost of trusted combiner required during signature generation phase. Therefore, the proposed scheme can be used as potential candidate for implimentation in proxy signature applications.

REFERENCES

[1] M. Mambo, K. Usuda and E. Okamoto , " Proxy signature: delegation of power to sign message" IEICE Transaction on Fundamental, vol E79-A, pp.1338-1353, 1996.

[2] S. Kim, S. Part and D. Won, " Proxy signature, revisited,"International Conference on Information and Communication Security ICICS'97, China Beijing, pp. 223-232, 1997.

[3] T. Elgamal, " A public key cryptosystem and signature scheme based on discrete logarithms," IEEE Transaction on Information Theory, Vo1.31, pp. 469-472, 1985.

[4] C. Schnoor, " Efficient signature generation by smart card," Cryptography, vol.4, pp. 161-174, 1991.

[5] K. Zang, " Threshold proxy signature schemes", Information Security Workshop ISW97, California, USA, pp.282-290, 1997.

[6] H. M. Sun, " Threshold proxy signatures," Computer and Digital Techniques, vol. 146, pp.259-263, IEE preceedings, 1999.

[7] M. S. Hwang, I. C. Lin and J. L. Lu," A secure nonrepudiable threshold proxy signature scheme with known signers", International Journal Informatica, Vol. 11, pp. 1-8, 2000.

[8] Z. Shao, " Improved threshold proxy signature schemes", Computer Standards and Interfaces, vol. 27, pp. 53-59 , 2004.

[9] M. S. Hwang, J. L. Lu and I. C. Lin, " A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, pp.1552-1560. 2003.

[10] G. Wang, F. Bao, J. Zhou and R. H. Deng," Comments on a practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp.1309-1311, 2004.

[11] Y. J. Geng, H. Tian and F. Hong, " A modified and practical threshold proxy signature scheme based on RSA", *Advance Communication Technology, 9th conference* Vol. 3, pp. 1958-1960, 2014.

[12] Y. Zhang, D. W. Yue and H. Zhang, " An improved (t, n) threshold proxy signature scheme with fault tolerance based on RSA", JICIC, Vol 6, pp. 3205-3218 2010.

[13] Z. W. Tan and Z. J. Liu, " Cryptanalysis and Improvement on a Threshold Proxy Signature Scheme" Journal of Information Science and Engineering, vol. 25, pp. 619-631 2009.

[14] Samaneh Mashhadi, "A Novel Non-repudiable Threshold Proxy Signature Scheme with Known Signers", International Journal of Network Security, Vol.15, No.4, pp.274-279, July 2013.

[15] Raman Kumar, Harsh Kumar Verma and Renu Dhir, " Analysis and Design of Protocol for Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers" Wireless Personal Communications, vol. 80 pp. 1281-1345. 2014.

[16] R.L. Rivest, A. Shamir and L.M. Adleman," A method for obtaining digital signatures and public-key cryptography" Communication of the ACM, Vol. 21, pp. 120-126, 1978.

[17] D. Boneh and Matthew Franklin. ,"Efficient generation of shared RSA keys,", Advances in Cryptography-CRYPTO '97, Springer- Verlag 1233 pp: 425-439, 1997.

[18] A. Shamir, "How to share a secret?" Communication of the ACM, Vol. 22, pp. 612-613, 1979.

[19] Ben-Or, M. Goldwasser and Wigderson , " Completeness theorems for noncryptographic fault-tolerant distributed computation," Proceeding of the 20th Annual ACM Symposium on Theory of Computing Chicago, I11, May 2-4, Newyork, pp. 1-10, 1998.

[20] H. L. Hguyen," RSA threshold cryptography" Dept. of Computer Science, University of Bristol, UK, 2005

[21] N. Koblitz, " A Course in Number Theory and Cryptography", Springer-Verlag, 1994.

 ISBN: 978-1-61208-427-5

The Use of Acceptance Test-Driven Development in the Construction of Cryptographic Software

Alexandre Melo Braga[12], Daniela Castilho Schwab[1], and André Luiz Vannucci[1]
[1] Centro de Pesquisa e Desenvolvimento em Telecomunicações (Fundação CPqD)
Campinas, São Paulo, Brazil
[2] Universidade Estadual de Campinas (UNICAMP)
Campinas, São Paulo, Brazil
Email:{ambraga,dschwab,vannucci}@cpqd.com.br

***Abstract*—This paper describes a work in progress on the usage of Acceptance Test-Driven Development (ATDD) during the construction of cryptographic software. As cryptography becomes universalized, it is becoming hard to separate good implementation from bad ones. The paper argues that Test Vectors for cryptography can be used as User Stories in Behavior-Driven Development (BDD) and automate ATDD during software development, complementing algorithm's specification, and contributing to augment software reliability and the overall trust in the correctness of cryptographic implementations. The acquired confidence is preserved even after performing program transformations for improvements, such as performance optimization and hardenings.**

Keywords-BDD; TDD; ATDD; User Stories; Security; Test vectors; Cryptography; Assurance.

I. INTRODUCTION

Nowadays, it is a well-accepted idea that the most likely attacks over software-based cryptosystems are against implementation faults and key management failures [2][16][17]. On the other hand, as cryptography becomes universalized, it is becoming difficult to assure that what is implemented in the real world is actually good cryptography.

The objective of this paper is to discuss preliminary results on the understanding of how the concepts of Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD) can be applied to the construction of cryptographic software, and how these two concepts can increase both the security and overall trust of software that rely on cryptographic implementations. The idea discussed here was experienced during the construction of a cryptographic library for Android devices [1].

Test-Driven Development (TDD) has become very popular in the agile programming community. In any secure software construction, the correctness of basic security functions is of major concern, and must be preserved even in an ever changing environment. The idea behind this text is that, in cryptography implementation, Test Vectors are User Stories formulated as automated acceptance tests, which can be successfully used to validate the implementation of a cryptographic algorithm against a specification, and prepare the room for future code optimizations and hardenings.

Once automated acceptance tests are available for a specification-based implementation of cryptography, further improvements on the source code can take place in order to address industry concerns, such as performance optimizations, power consumption, and security controls against side-channel attacks and other vulnerabilities in source code. Even after all these transformations, acceptance tests preserve trust by giving strong evidence of correctness.

This work has two motivating drivers. The first is an actual need for increasing the confidence in cryptographic algorithm implementations, which are not under the scrutiny of cryptologists. It is a fact of life that cryptologists are scarcely available human resources, and ordinary programmers are not only more available, but less expensive as well. A frequently asked question in industry is whether or not it is possible to produce high quality implementations of good cryptography, even when there is no cryptologist neither writing source code nor deeply inspecting it.

The second is a lack of literature concerning the combination of TDD, ATDD, or BDD and specific security technologies, like cryptography. Today, common usages of TDD are related to general topics in software, such as enterprise applications, mobile code, data-access code, and so on. The authors could not find any reported case of TDD in cryptographic software.

The text is organized as follows. Section II offers background information on related subjects. Section III details the proposed idea. Section IV discusses practical issues of the proposed approach. Section V contains concluding remarks and future work.

II. BACKGROUND AND RELATED WORK

This section offers background on ATDD, BDD, TDD for security, and Test Vectors for cryptography validation.

A. Acceptance Test-Driven Development

ATDD is strongly related to BDD, both of them drive TDD, Acceptance Tests, and Unit Test from User Stories. The core idea of TDD was proposed in 2002 [7] and the state of the practice was studied recently [18], with good text books available on the subject [9].

ATDD drives development on the feature level, similarly to TDD in code level with Unit Tests. Acceptance tests act as micro specifications for the desired behavior and functionality of a system. They tell how the system handles certain conditions and inputs and with what kinds of consequences and outputs. The benefits of ATDD are the

 ISBN: 978-1-61208-427-5

following: (i) clear definition of "work done", by providing the knowledge of where the development is and of when to stop working; (ii) promotion of trust and commitment, because there's a direct connection between what the customer specifies and what she gets; and (iii) specification by example, when requirements are expressed by comprehensible examples, rather than by complex formulas or ambiguous descriptions. Tests expressed with concrete examples are easier to read, easier to understand, easier to validate, and easier to write [9].

In Acceptance TDD, a requirement is translated into a set of executable tests and then applied to the implementation, which is validated against tests, rather than against the developer's interpretation of a requirement.

1) User Stories

User Stories are a useful, lightweight technique for managing requirements. User Stories are short sentences written with customer's assistance, stating who does what and why. The story is intended to represent a requirement, acting as a promise of a future conversation between the customer and the developer. A story is typically only one sentence long, it is not intended to document the requirement, and it does not substitute actual specifications.

The most common format or template for User Stories contains the name of the story and three phrases: As a *[user role of the system]*, so that *[I can achieve some goal or objective]*, I want to *[perform some task]*. These three phrases resemble a simple desire of users or customers.

Ideally, a User Story can be formulated as acceptance test before code is written. Well written User Stories, that produce good acceptance tests, usually have quality attributes called Specific, Measurable, Achievable, Relevant, and Time boxed (SMART). The SMART attributes mean that it has to be at least a pair of valid input and corresponding output that: (i) is expressed in the language of the domain specialists; (ii) is concise, precise and unambiguous; and (iii) could be tested within a finite (and short) amount of time. As discussed further in this text, cryptographic test vectors comply with all these attributes.

2) Behavior-Driven Design and TDD

Behavior-Driven Design (BDD) is a way to develop User Stories to describe features on computer programs. BDD concentrates on program's behavior, instead of its implementation. In BDD, the development team asks questions about program's behavior, before and during development, to reduce miscommunication. The questions generate requirements written down as simple User Stories. Later on, User Stories become acceptance tests and integration tests of those programs.

The advantages of BDD are that User Stories are expressed in common language for all stakeholders, and make it feasible to write tests before or during coding. This only feature turns debugging time into validation time. The disadvantages of BDD are twofold: (i) continuous contact with customer is difficult to achieve in most software projects and (ii) BDD almost always leads to bad software architecture, thus requiring frequent refactoring of source code. The compliance to standard Application Programming Interfaces (APIs) and algorithm specifications minimize the impact of this disadvantage in cryptographic software.

Test-Driven Development (TDD), or Test-First Development (TFD), is the practice of writing automated Unit Tests for low-level program constructions (e.g., objects) based on simple User Stories. TDD is guided by a sequence of User Stories obtained from the customer or user. On TDD, the supposed result of writing low-level Unit Tests is that only few defects show up during tests.

Advocates of TDD may question the usefulness of Unit Tests in the presence of automated ATDD. Unit Tests are still useful to validate compliance to programming contracts of an API or to the programming dialog of Frameworks, contributing to regression tests when acceptance tests are not effective. That is exactly the case of cryptography implementation, when testing accessory functionality, such as padding schemes, and conformance to APIs are requirements.

B. TDD and Software Security

There are few works relating TDD or ATDD and security [3][10][21][22]. The work of Smith, Williams, and Austin [3] assesses the relative effectiveness of system and unit level testing of web applications to reveal both SQL injection vulnerabilities and error message information leakage vulnerabilities, when used with an iterative test automation practice by a development team.

More recently, three related works [22][21][10] addressed TDD for security testing. First, Kobashi et al [22] proposed a method to validate implementations of security pattern using TDD. In this method, developers specify the threats and vulnerabilities in the target system during an early stage of development, and then the proposed method validates whether the security patterns were properly applied and assessed whether vulnerabilities were resolved.

Then, Yoshizawa et al [10] evolved the previous work by proposing a validation method, using TDD, for security design patterns in the implementation phase of software development. Finally, Kobashi et al [21] implemented their method in a tool called TESEM (Test Driven Secure Modeling Tool), which supports pattern applications by creating a script to execute model testing automatically. During an early development stage, the developer specifies threats and vulnerabilities in the target system, and then TESEM verifies whether the security patterns are properly applied and assesses whether vulnerabilities are resolved.

None of the above mentioned works treat ATDD in the context of cryptographic software development.

C. Test Vectors for Cryptography

Test Vectors have been used in validation of cryptographic implementations for many years, mostly for product certification, post construction. This section describes the validation of cryptographic implementations with Test Vectors during development.

Test Vectors are data sets constructed with the aim of evaluating the correctness of cryptographic implementations, not their security. However, the functional correctness is a

 ISBN: 978-1-61208-427-5

strong prerequisite for security because, in principle, an incorrect implementation is both unreliable and insecure.

In order to make de validation feasible, cryptographic software under evaluation should allow the necessary control over the input parameters needed for testing. For example, the ability to configure or load known values for the variables required for a specific test may be available via an API. Tests cannot be performed if cryptographic software does not allow control over the values of input parameters.

There are publicly available Test Vectors [11][13][14]. A well-known set of vectors is provided by US National Institute of Standards and Technology (NIST) within the Cryptographic Algorithm Validation Program (CAVP) [13]. All validations based on Test Vectors are designed to test compliance with the norms and standards of the specific algorithm being evaluated. Therefore, they are not meant to provide a measure of the security for a particular cryptographic implementation.

Crafted validation tests are designed to detect accidental defects of implementation and operation, and are not designed to detect intentional attempts to misrepresent validation. For example, malicious implementations can be constructed to give the correct answer for a particular set of tests, then passing as a correct implementation, while concealing some other malicious function. Hence, it is a good practice the use of updated, randomly-generated vectors in conjunction with crafted or standard vectors.

It is noteworthy that Test Vectors are constructed using statistical sampling. That is, only a small amount of samples is extracted from the universe of test cases. Therefore, the successful validation implies strong evidence, but not absolute certainty, of correctness for the implementation under evaluation.

In order to exemplify the structure of Test Vectors, this text uses the Advanced Encryption Standard (AES) [12], along with NIST's vectors [8]. The validation of AES covers various operation modes (e.g., ECB, CBC, OFB, CFB1, CFB8, and CFB128). For each mode, three key sizes are selected (128, 192, and 256 bits).

The AES validation consists of three types of test: Known Answer Tests (KAT), Multi-block Message Test (MMT), and Monte Carlo Test (MCT). There are extra vectors for GCM and XTS modes. The KAT test suite tests four algorithm-specific components. For instance, the GFSbox set tests finite field arithmetic, the KeySbox set tests transactions on subkeys, the Variable Key set tests fixed plaintext against varying keys, and the Variable Text set tests fixed keys against varying plaintext or cipher texts.

```
01 @Test
02 public void pkcs5PaddingTest(){
03    byte[] result = new byte[8],
04    input = "AAAA".getBytes();
05    int inputOffset=0, inputLen=4;
06    int totalInputLen=8, blockLength=8;
07    result = Padding.pkcs5Padding(input,
08          inputOffset,inputLen,totalInputLen,
09          blockLength);
10    assertEquals(Util.ByteArrayToHexStr(result),
11                                 "4141414104040404");
12 }
```

Figure 1. Unit Test for PKCS#5 padding of size 4 on a 8-byte block.

TABLE I. FOUR AES KAT VECTORS (ENCRYPTION, CBC, 128-BIT KEY).

Vector type and index	Vector value for each parameter
GFSbox test data for CBC #0	KEY = 00000000000000000000000000000000 IV = 00000000000000000000000000000000 PT = f34481ec3cc627bacd5dc3fb08f273e6 CT = 0336763e966d92595a567cc9ce537f5e
VarKey test data for CBC #0	KEY = 80000000000000000000000000000000 IV = 00000000000000000000000000000000 PT = 00000000000000000000000000000000 CT = 0edd33d3c621e546455bd8ba1418bec8
KeySbox test data for CBC #0	KEY = 10a58869d74be5a374cf867cfb473859 IV = 00000000000000000000000000000000 PT = 00000000000000000000000000000000 CT = 6d251e6944b051e04eaa6fb4dbf78465
VarTxt test data for CBC #0	KEY = 00000000000000000000000000000000 IV = 00000000000000000000000000000000 PT = 80000000000000000000000000000000 CT = 3ad78e726c1ec02b7ebfe92b23d9ec34

The MMT tests are designed to test the ability of the implementation to process input data consisting of many blocks, and require correct implementation of chaining from block to block. Both KAT and MMT are simple comparisons of known values. MCT still performs comparisons of known values of ciphertexts, but the current ciphertext is computed by chaining previously generated ciphertexts as input to new encryptions into a loop. The last ciphertext is then compared to the value of test vector.

Table I shows KAT vectors for AES encryption in CBC mode with a 128-bit key. The table follows NIST's format and contains examples of the four KAT subtypes. IV, PT and CT stand for Initialization Vector, Plain Text, and Cipher Text, respectively.

III. TDD AND ATDD FOR CRYPTOGRAPHY

This section proposes an approach to perform TDD and ATDD over cryptographic implementations. First, it discusses a strategy for conducting Unit Tests that fits on the TDD framework. Then, a strategy to perform ATDD with Test Vectors as User Stories is presented and discussed.

The code snippet in Figure 1 is an example of how JUnit [6], a simple framework to write repeatable tests in Java, can be used in automated testing of security functions. The method tests the structure of padding in PKCS#5 format. The code works for blocks of 8 bytes (for example, used by 3DES) and the input data of 4 bytes, so the function must include 4 bytes of padding with the hexadecimal value 0x4. This code can be generalized to other test cases for padding.

In the case of padding, the exhaustive coverage of all possible test cases becomes feasible, since the padding has a small number of options that depend on the block size of the cryptographic algorithm. For instance, there are 8 test cases for algorithms with 64-bit block, as well as 16 test cases for

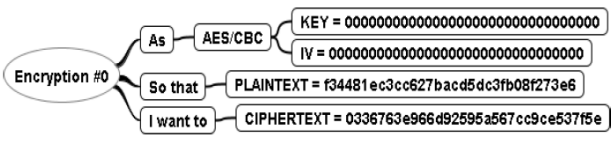


Figure 2. The User Story for a single Test Vector.

algorithms with 128-bit blocks. Considering both insertion and removal of padding as distinct test cases, there are 48 test cases. In terms of JUnit, there are two approaches for implementing these test cases. In one, all the 48 test cases can be individually automated. In other, one single function can be built for all test cases. An option sitting in between is to build two separated functions, one for padding additions and other for removals.

The choice of a method for each test case is preferable because complies with TDD's philosophy of identifying errors quickly and directly. Moreover, the option with a single method encapsulating all test cases still implies that there must be debugging to identify which test cases have failed. Therefore, by not eliminating depuration, it yields only partial benefit from TDD philosophy. A similar approach can be adopted to test cryptographic routines, but with major limitations. TDD and JUnit can be used to test the basic operation of the encryption routine by using particular Test Vectors for encryption and decryption.

Despite being useful for simple functional tests of security functions, unit tests (based upon JUnit) do not scale well when applied to ATDD for cryptography. If each possible cipher text is considered a test case, then the amount of test cases, though finite and countable, is incredibly large. Even a small sample, but statistically significant, greatly increases the time to perform unit tests. For this reason, ATDD is most suitable for the validation of cryptographic implementations than simple TDD based upon stand-alone JUnit tests. Figure 2 illustrates the first vector of Table 1 represented as a User Story for AES encryption. This single test case is one in thousands of test cases. For instance, NIST's vectors for AES in CBC mode consist of more than 2,700 single tests.

Figure 3 depicts the overall idea of using ATTD for cryptography. Test vectors are input data to test cases, which are programmed as (automated) User Stories intended for Acceptance Tests. Then, ATDD asserts the expected behavior of cryptographic algorithms, resembling BDD. This approach can be complemented by ordinary unit tests. The point of contact between User Stories and cryptographic implementations is the cryptographic API. In this way, test cases do not act directly on algorithms internals, but verifies its behavior, as seen from the API perspective.

Java programs, such as the one shown in Figure 4, were built to enable ATDD through Java Cryptographic Architecture (JCA) [4][5]. Before being used in a custom cryptographic library [1], the ATDD test suite was validated against two presumed correct implementations of JCA [5][23]. The test suite was used for testing not only pure Java code, but also C code encapsulated by Java Native Interface (JNI) adapters and available through the JCA API.

Figure 4 illustrated how NIST's Monte Carlo Tests (MCT) [8] can be performed by the proposed ATDD test suite. The figure shows the source code for a Java method to perform MCT tests for encryption in ECB mode. This Java code is almost a direct translation of the pseudo code from [8]. This code snippet was meant for AES, but can be generalized for any block cipher, because it does not depend on the specific test data nor cipher implementation. The

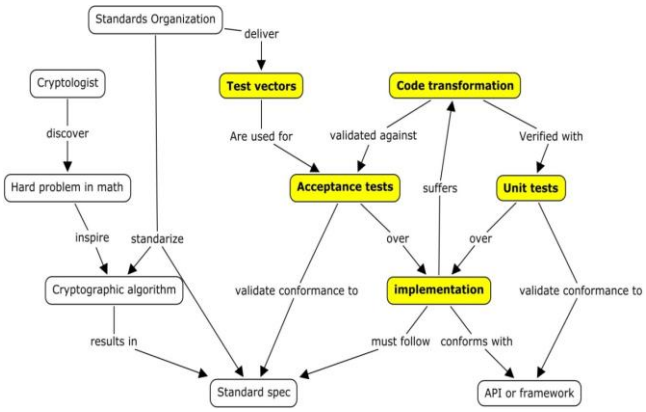


Figure 3. ATTD asserts whether an implementation follows its specified behavior.

cipher function (a wrapper for the actual encryption function) is called only two times, in lines 17 and 19. The loop from line 10 to line 36 computes a chain of ciphertexts, which is saved (in line 25) for future comparisons.

IV. PRACTICAL ISSUES AND DISCUSSION

This section discusses practical considerations that arise when implementing the proposed approach.

```
01  void mcEncECB(byte[] key, byte[] plainText) {
02    byte[][] bK  = new byte[1000][];//WorkingKey
03    byte[][] bPT = new byte[1000][];//PlainText
04    byte[][] bCT = new byte[1000][];//CipherText
05    TestVector vt;
06    calcVT = new TestVector[100];
07    bK[0] = key;
08    bPT[0] = plainText;
09
10    for (int i = 0; i <= 99; i++) {
11      vt = new TestVector();
12      vt.setPT(b2x(bPT[0]));
13      vt.setKey(b2x(bK[i]));
14      int j;
15      for (j = 0; j <= 999; j++) {
16        if (j == 0) {//init cipher with key
17          bCT[j] = crypt(bK[i], bPT[j], true);
18        } else      {//reuse cipher
19          bCT[j] = crypt(bK[i], bPT[j], false);
20        }
21        if (j < 999) { bPT[j+1] = bCT[j];}
22      }
23      j--; // leaves loop when (j == 1000)
24      vt.setCT(b2x(bCT[j]));
25      calcVT[i] = vt;
26      if (i <= 99) {
27        if (key.length == 16) { // 16*8 = 128 bits
28          bK[i+1] = xor128(bK[i], bCT[j]);
29        }
30        if (key.length == 24) { // 24*8 = 192 bits
31          bK[i+1] = xor192(bK[i], bCT[j-1], bCT[j]);
32        }
33        if (key.length == 32) { // 32*8 = 256 bits
34          bK[i+1] = xor256(bK[i], bCT[j-1], bCT[j]);
35        }
36      }
37      bPT[0] = bCT[j];
38    }
39    return;
40  }
```

Figure 4. Monte Carlo Test for encryption in ECB mode, in Java.

 ISBN: 978-1-61208-427-5

A. Examples of Defects Found

Defects did occur during development. Thus, in addition to assert compliance to specifications, test cases are structured to detect implementation faults, including problems with pointers, insufficient memory allocation, incorrect treatment of errors, and other incorrect behaviors. This section gives examples of the more interesting defects found, usually not related to simple misunderstandings of specifications.

The proposed method was applied over standard (e.g., AES, RSA, HMAC, and SHA-2) and non-standard (e.g., Serpent, Salsa20, and Blake) cryptography. When testing the implementation of AES, two kinds of failures were identified. Failures in padding were found when zero padding and AnsiX923 padding were used and the value of the last byte was less than 16, resulting in a mismatch in test vectors. A failure in the CTR mode of operation occurred when the first 16 bytes (first block) was correctly encrypted (matched test vector), but the remaining blocks do not. Other failure was caused by a missing initialization of the hash function after the first call to it into HMAC computation.

Non-standard algorithms produced more severe failures than standard ones. It is worth to mention the cases for three algorithms: Serpent, Blake, and Salsa20. When testing a C implementation of Serpent against its test vectors [19], several implementation failures were discovered concerning memory leaks, wrong output for either encryption and decryption, and program crashes after 1,000 iterations.

Salsa20 and Blake lack extensive test vectors. In fact, only a few unofficial tests were found for Salsa20, which were complemented by random tests produced by a reference implementation. Blake was tested only with random tests produced by a reference implementation. A bug was detected in Blake that was caused by wrong calls from Blake224 to functions of Blake256. Also, platform upgrades (from 32 bits to 64 bits), downgrades (from 64 to 32 bits) and changes (from different flavors of Linux) caused errors in encryption and hashes that could be detected by regression tests.

Similar defects were found during development of all cryptographic algorithm implementations. By the time of writing, there were no collected statistics on the efficiency of the proposed method. However, the examples mentioned above suggest the method worth the effort when there is no cryptologist timely available to support debugging.

B. Lessons Learned

Standard algorithms produced fewer defects than non-standard ones. That is probably because standard algorithms possess better documentation available to developers as well as good reference implementations.

The Java APIs for symmetric encryption, secure hash functions, and Message Authentication Codes (MAC) have shown good testability (meaning the ease with which a given test coverage criterion can be satisfied [15]) against NIST's vectors. Unfortunately, the same cannot be said for asymmetric encryption. JCA was created before the advent of TDD and ATDD, and is not testable by design. The authors have found that parts of the API presented poor testability and are not suitable for testing with NIST's vectors. In particular, the API for asymmetric encryption was designed with "textbook" RSA in mind, and does not allow for RSA-PSS and RSA-OAEP to be easily tested, because it's not possible to setup some of the parameters required by NIST for these two randomized algorithms. Similarly, the key agreement API was designed with "textbook" Diffie-Hellman (DH) in mind and suffers from the same issue when used with Authenticated DH implemented according to NIST's specifications. This means that, in order to be fully testable, an implementation has to sacrifice conformance to JCA API. Also, security issues have been found in JCA [16].

Concerning API compliance, a lesson learned is that any cryptographic implementation should have the ability of being tested by third parties. Co-design of both functional code and test code can favor testability. But that may not the case when testing legacy cryptography with ATDD.

TDD and ATDD can in fact reduce the time of debugging, when looking for the causes of failures. A lesson learned is that TDD uses the same techniques of debugging, but in a productive way, writing automated tests during software development.

Failure isolation seems to be the most important advantage of TDD to development of cryptographic software. In TDD, if a test fails, the cause must be the most recently added code. Failure isolation is almost trivial in TDD, because at any moment, all previous test cases must have passed. Modes of operation as well as the internals of cryptographic implementations are hard to debug otherwise.

Usually, good test cases are not sufficient for TDD to produce good code. Also, good design is eventually accomplished by code refactoring. In the case of standard cryptographic algorithms, there will always be specifications and reference implementations. Furthermore, conformance to an API (e.g., JCA), minimizes the need for refactorization.

Finally, the most criticized disadvantage of TDD is that it is strongly dependent on tester's experience to produce good test cases. In cryptography, there is a concern about the need for an oracle that could provide good-enough test cases. According to [15], an oracle is any (human or mechanical) agent that decides whether a program behaved correctly in a given test and accordingly gives a verdict of pass or fail. In case of cryptography, that need is satisfied by test cases provided either by cryptologists or standards organizations.

Unfortunately, non-standard algorithms usually lack test cases and can only benefit from relatively small test sets supplied by their authors or other practitioners. In this work, for both standard and non-standard cryptography, reference implementations were used as oracles for generation of random test cases, in complement of third-part test vectors.

C. Test Vectors as Metrics for Quality Measures

In order to be useful as a quality measure, tests must have clearly defined meanings for success and failure. This text adopted the meanings from the Software Engineering Body of Knowledge (SWEBOK) [15] as follows: a fault is the (root) cause of a malfunction and a failure is the undesired effect observed in the behavior of programs. Thus, testing can reveal failures, but it is the faults that can and must be removed from programs. Still [15], the generic term defect

 ISBN: 978-1-61208-427-5

can refer to either a fault or a failure. The result of testing a single cryptographic implementation with its test vectors is twofold: passing them all constitutes success with a justified confidence, but failing only once reveals a failure and compromises the whole implementation.

Failed test cases can be used as indirect measures of how distant an implementation is from achieve conformance to that test cases and can be used for estimation of effort, team assignment, overall cost estimation, and influencing how long a test effort should be continued. In this work, the criteria for test termination could be positively defined by passing all test cases. The term "passed them all" was directly related to how much testing was enough and when a test period could be concluded. It also involved concerns about costs and risks incurred by possible remaining failures, as opposed to costs incurred by continuing to test.

Additionally, when considering whole cryptographic libraries with implementations for various algorithms, to make testing more effective in making quality predictions, it is important to know which types of failures may be found and the relative frequency with which these failures have occurred in the past. The failure density for an implementation under test can be evaluated by counting discovered failures as the ratio between the number of failures found and the size of that implementation. This evaluation was left as a future work.

V. CONCLUDING REMARKS

This paper argues that Test Vectors are User Stories and can automate acceptance tests for cryptographic software. Test Vectors are good acceptance tests because they meet halfway between cryptologists and developers. They are User Stories from the problem domain, that don't look like source code, providing an easy way to reach agreement. The approach presented in this text increases confidence in cryptographic software by maintaining a strong evidence of correctness, even after many code transformations.

Future work includes the use of ATDD in other cryptographic algorithms and protocols. Further research includes the design of customized Test Vectors. In order to be useful, metrics and statistics concerning the efficiency of the approach still have to be collected in structured ways. Finally, studies have to be done to combine the approach with methods for secure software development.

ACKNOWLEDGMENT

The authors acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11.0028.00 with the Financier of Studies and Projects - FINEP/MCTI.

REFERENCES

[1] A. Braga and E. Morais, "Implementation Issues in the Construction of Standard and Non-Standard Cryptography on Android Devices," in proc. of the 8th International Conf. on Emerging Security Information, Systems and Technologies (SECURWARE), 2014, pp. 144–150.

[2] B. Schneier, Security in the Real World: How to Evaluate Security Technology, Comp. Sec. Journal, 1999, vol.15, n. 4.

[3] B. Smith, L. Williams, and A. Austin, "Idea: using system level testing for revealing SQL injection-related error message information leaks," *in* Proc. of the 2nd International Conference on Engineering Secure Software and Systems (ESSoS), 2010, pp. 192-200.

[4] Java Cryptography Architecture (JCA) Reference Guide. [retrieved: July, 2015] docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html.

[5] Java Cryptography Architecture, Providers Documentation for JavaSE 7. [retrieved: July, 2015] docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html.

[6] JUnit 4. [retrieved: July, 2015] http://junit.org.

[7] K. Beck, Test Driven Development By Example, Addison-Wesley Longman, 2002.

[8] L. E. Bassham III. The Advanced Encryption Standard Algorithm Validation Suite (AESAVS). National Institute of Standards and Technology, 2002.

[9] L. Koskela, Test Driven: Practical TDD and Acceptance TDD for Java Developers, 2007.

[10] M. Yoshizawa, T. Kobashi, H. Washizaki, Y. Fukazawa, T. Okubo, H. Kaiya, and N. Yoshioka, "Verifying Implementation of Security Design Patterns Using a Test Template," in proc. of the 9th Int. Conf. on Availability, Reliability and Security (ARES), 2014, pp. 178–183.

[11] NESSIE Test Vectors. [retrieved: July, 2015] www.cosic.esat.kuleuven.be/nessie/testvectors.

[12] NIST FIPS-PUB-197, Announcing the Advanced Encryption Standard (AES), FIPS Publication 197, 2001.

[13] NIST. Cryptographic Algorithm Validation Program (CAVP). [retrieved: July, 2015] csrc.nist.gov/groups/STM/cavp.

[14] OpenSSL Validation Suite. [retrieved: July, 2015] opensslfoundation.com/testing/validation-2.0/testvectors.

[15] P. Bourque and R. Fairley, Eds., Guide to the Software Engineering Body of Knowledge (SWEBOK), ver 3. IEEE Comp. Society, 2014.

[16] P. Gutmann, "Lessons Learned in Implementing and Deploying Crypto Software," in Proc. of the 11th USENIX Security Symposium, Dan Boneh (Ed.), 2002, pp. 315-325.

[17] R. J. Anderson, "Why cryptosystems fail," Communications of the ACM, vol. 37, n. 11, 1994, pp.32-40.

[18] S. Hammond and D. Umphress, "Test driven development: the state of the practice," in proc. of the 50th Annual Southeast Regional Conf., 2012, pp. 158-163.

[19] Serpent - A New Block Cipher Proposal for AES. [retrieved: July, 2015] www.cs.technion.ac.il/~biham/Reports/ Serpent.

[20] Snuffle 2005: the Salsa20 encryption function. [retrieved: July, 2015] http://cr.yp.to/salsa20.html.

[21] T. Kobashi, M. Yoshizawa, H. Washizaki, Y. Fukazawa, N. Yoshioka, T. Okubo, and H. Kaiya, "TESEM: A Tool for Verifying Security Design Pattern Applications by Model Testing," in proc. of the IEEE 8th Int. Conf. on Software Testing, Verification and Validation (ICST) , 2015, pp. 1–8.

[22] T. Kobashi, N. Yoshioka, T. Okubo, H. Kaiya, H. Washizaki, and Y. Fukazawa, "Validating Security Design Patterns Application Using Model Testing," in proc. of 8th Int. Conf. on Avail., Reliability and Security (ARES), 2013, pp. 62–71.

[23] The Legion of the Bouncy Castle. [retrieved: July, 2015] www.bouncycastle.org/java.html.

 ISBN: 978-1-61208-427-5

Secret Sharing Schemes Threshold Determination

Armindo Guerra Jr.

Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil
Email: armindogjr@gmail.com

Ricardo Felipe Custodio

Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil
Email: custodio@inf.ufsc.br

***Abstract*—In threshold secret sharing schemes, the threshold and the total number of shareholders are public information. We believe that such information should be secret and the threshold value must be determined before the secret reconstruction. Thus, this paper propose a method to do it. We propose also, an analysis of tailored access structure that has different subsets to rebuild the secret, each subset with his threshold. Furthermore, in our investigation we have seen that is possible that there are malicious access structures where a previleged subgroup of shareholders, in smaller number than the threshold, can reconstruct the secret. Finally, we propose a method to choose a polynomial that does not generate malicious access structures.**

Keywords–Secret sharing; threshold; information

I. INTRODUCTION

Secret sharing schemes have been used to protect sensitive data through its division into pieces and subsequent distribution to various different custodians, with the possibility that a portion of these shares can be used to recover the original data. The first known constructions that allow to share a secret were proposed independently by Shamir [1] and Blakley [2]. Shamir's scheme uses polynomial interpolation to recover the secret, while Blakley's scheme is based on the geometric intersection of hyperplanes. Many other schemes have been proposed since then and we can highlight the schemes of [3] [4], which propose general constructions and Mignotte's scheme [5], which is based on the Chinese remainder theorem. A good summary of secret sharing schemes can be found in Beimel's survey [6].

According to [7], secret sharing scheme allows a Dealer (D) to protect a secret S among a set of n shareholders. The access structure of the scheme is the set of subsets of the shareholders that are able to reconstruct the secret using their shares. A special case called (t, n)-threshold access structure consists of subsets containing at least t shareholders, where $1 < t \leq n$. In this case, any t or more shares out of n shareholders can recover the secret.

In (t, n)-threshold secret sharing schemes, t and n are public information, and, furthermore, a possible reconstructor (R) needs to know how many t shares are needed to reconstruct the secret. In this scenario, an attacker knows how many shareholders need to be coerced in order to reconstruct the secret by himself. This paper proposes that the value of t do not must be published. Therefore, our main contribution is to show that it is possible to reconstruct the secret without first knowing the threshold.

A secret sharing scheme begins with a secret and derives from it certain number of shares. The secret must be reconstruct only by certain predetermined subsets of shareholders. Let $\mathcal{V}$ be the set of n shareholders and $\mathcal{B} = 2^{\mathcal{V}}$ be the power set of $\mathcal{V}$. The set $\mathcal{B}$ is partitioned into 2 sets, $\mathcal{B} = \overline{\Gamma} \cup \Gamma$, where $\overline{\Gamma}$ is the complement of Γ. We call Γ the access structure; it contais all the subsets of $\mathcal{B}$ with cardinality greater than or equal to t. Thus, any element $x \in \Gamma$ can be used to rebuild the secret and any $x \in \overline{\Gamma}$ cannot do it. The D is the entity responsible for spliting the secret into n shares and for delivering each share to the shareholders. The R is the entity responsible for collecting shares and performs the reconstruction of the secret.

Due to our method, to reconstruct the secret without first knowing the threshold, we detected that is possible more than one access structure to reconstruct the same secret. Different subsets of shareholders will have different thresholds. Furthermore, we realize that is possible that there are malicious access structures where a previleged subgroup of shareholders can reconstruct the secret. thus, for an access structure be reliable it is necessary avoid such previleged subgroups of shareholders. Hence, we also propose a method to do it.

This document is organized as a follows. In Section II, we present a method to determine the threshold before recontructing the secret and its implications. In Section IV, we present some considerations about the propose methods. Finally in Section V, we present our conclusions and future works.

II. RECONSTRUCTING THE SECRET WITHOUT t AND n

In threshold secret sharing schemes, we are interested in building an (t, n)-access structure, where t is the threshold and n is the total number of shareholders. With at least t shares, we can determine the secret S. Normally, the secret S is the independent term of the interpolating polynomial and both t and n are public values.

In this work, however, we argue that these values should be secret. In fact, R do not need to know t nor n in order to determine the secret S. Thus, R needs a way of knowing he has used the minimum number of shares necessary to obtain the correct interpolating polynomial and so the secret S.

A. Threshold determination method

We suppose that D divides a secret S into n shares using a polynomial whose degree is $t-1$. Now, let's suppose that R wants to obtain the secret S from the w shares. As R does not kown t, then in principle it cannot reconstruct S. Nevertheless,

 ISBN: 978-1-61208-427-5

we supose also that R perform an interpolation of w shares and d is the degree of interpoling polynomial. It is easy to check that if $d < w-1$, then $t = d$. Thus, R has more shares than necessary to reconstruct S. However, if $d \geq (w-1)$ R cannot determine t. Therefore, R just get many shares until the degree of the polynomial is $d = w-1$.

Theorem 1: Let t and n be the threshold secret sharing scheme parameters, where t is the threshold and n is the number of total shares. Let w be any number of points able to participate of the reconstruction of the secret. It is possible to determine t if $w > t$.

Proof: To determine the threshold we interpolate the w shares and verify if the degree d of the interpolating polynomial is less than $w-1$. If so, $t = d+1$. If not, the w shares are not enough to recover the secret S. In this case, we must increment w and repeat the reasoning until the degree of the interpolator polynomial is equal to $w-2$. This only happens when $w > t$. ■

As R does not know t nor n, two approachs are possible. The first R has no cost to picking up shares arbitrarily from Γ. In this case it is possible to offer to R all n shares. then, R interpolates n shares and easily finds the value of t. This can be done according Algorithm 1. The second R has significative costs to picking up shares. Thus, it is interesting to search for the value of t starting with the minimum possible number of shares. This search can be done according Algorithm 2.

We know that the threshold t certainly assumes a value between 2 and n. Therefore the search begin with 2 shares, because are needed at least two shares to interpolate a polynomial of degree equal to 1. Nevertheless, the methodology proposed by this paper always use one more share than traditionally required. Thus, the search starts with 3, as can be seen in Algorithm 2. It is important to observe that the proposed method should not be used in unanimous secret sharing schemes, when $t = n$, because a redundant share is always necessary. We consider redundant share one more share than the required in tradicional threshold secret sharing schemes.

Algorithm 1: checkThreshold( )

```
Input: shareList (share = (x_i, P(x_i))), 1 < i < w
Output: threshold t or −1
1  shareList = [ ];
2  w ← shareList.length();
3  polynomial ← interpolate(shareList) ;
   // interpolate() is a method which
   returns a interpoling polynomial
4  degree ← getDegree(polynomial) ;
   // getDegree() is a method which
   returns a degree of polynomial
5  if degree ≤ w − 2 then
6  |  t ← degree + 1
7  else
8  |  t ← −1
9  return t
10
```

In the first strategy, when there is no cost to R shares picks up shares, only the Algorithm 1 is necessary.

Algorithm 2: Search threshold

```
Input: AllSharesList
Output: threshold t or −1
1  t ← −1;
2  AllSharesList =[ ];
3  for i ← 0 until 2 do
4  |  shareList.append(AllSharesList.extractShare( )) ;
   |  // extractShare() is a method which
   |  picks up and delete individually
   |  shares from AllSharesList[ ]
5  repeat
6  |  t ← checkThreshold (shareList);
7  |  if t ≠ −1 then
8  |  |  return t;
9  |  else
10 |  |  shareList.append(AllSharesList.extractShare( )) ;
   |  |  // One more share
11 until AllSharesList.length() = 0 ;
12 return t
```

B. False positives

The method presented in Section II-A, shows how to retrieve the secret S without knowing the threshold t. However, care must be taken with some consequences of using redundant to do it. As Algorithm 2 increases the number of shares picked up from the access structure in an ascending order, polynomials formed by the elements of the same access structure and whose degree is smaller than $t-1$ could be found. In this case the present method would return a false value of t. We call those false positives polynomial.

Definition: False positives polynomial are polynomial formed by the shares of the access structure whose degree is smaller than that of the polynomial interpolating.

As example, if we offer for Algorithm 1 the shares $\mathcal{A} = \{p1, \ldots, p8\}$, where $\mathcal{A} \subseteq \Gamma$, the result is polynomial $f(x) = 103/384x^6 - 3605/384x^5 + 24205/192x^4 - 154439/192x^3 + 308949/128x^2 - 361631/128x + 99$, each degree is 6. As we can see in Figure 1. However, when have significative costs to picking up shares the Algorithm 2 is used. In this case a polynomial $g(x)$ with degree is 4 will be found. Also we can see in Figure 1. In this case $g(x)$ is a false positive polynomial. It is a problem because a different polynomial can be to result in a diffente secret.

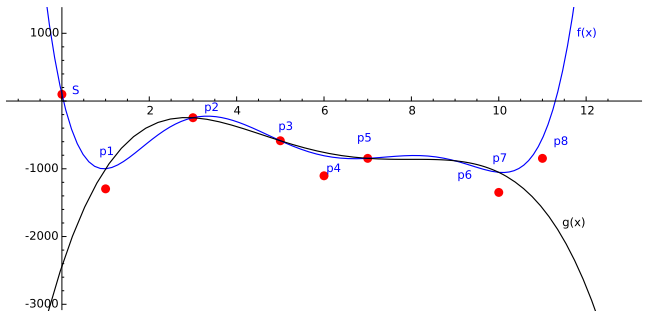


Figure 1. False positive

Knowing the possibility of false positives existence to use the method proposed by this work is necessary to generate a tailored access structure with no false positives. As the method to find false positives polynomial we propose the Algorithm 3. The objetive of this algorithm is to recieve the list of the shares from the access structure as the algorithm entry and as a result to give back all possible false positives polynomials.

Algorithm 3: Search of curves

Input: shareList
Output: A list possibleCurves, which return all possible polynomial formed using a redundant share

```
1 possibleCurves = [ ];
2 listLength ← shareList.length();
3 for j ← 3 until listLength do
4     combinationsList ← combinations(shareList,j) ;
      // combinations() is a method which
      returns the combinations of
      shareList's shares taken j by j
5     for i ← 0 until combinationsList.cardinality() do
          // combinationsList.cardinality()
             is a method whith returns the
             combinations of share of
             shareList taken j by j
6         q ← interpolate(combinationsList[i]);
7         if q.getDegree() = j − 2 then
8             possibleCurves.append(q);
9 return possibleCurves;
```

C. Generation of access structure

In the literature there are some ways to generate an access structure for a threshold secret sharing scheme. Iftene [8], for example, choose a interpolating polynomial with his coefficients over $\mathbb{F}[x]$ and choose the shares $s_1, \ldots, s_n$ as $s_i = P(x_i)$, for all $1 \leq i \leq n$, where $x_1, \ldots, x_n$ are pairwise distinct public values. In this study we consider that the access structure is formed by all ordered pairs values (x_i, s_i).

However, it is necessary to generate an access structure whose shares do not yeld false positive polynomials. A pratical way is to choose a random polynomial and test whether the access structure generated by it has or not false positives. If it has, we choose another random polynomial and perform the test again until we find a polynomial that has no false positives.

After choosing the interpolating polynomial, one way to test it if there are false positives is to give the shares generated by the interesting polynomial as the entry to the Algorithm 3. Basically the Algorithm 3 tests all combinations shares from the access structure and shows only the polynomial formed obeying the requirement of redundant share.

III. Consequences

A. Hidden access structures (HAS)

In our investigations about the tradition threshold secret sharing scheme, we discovered that there may be malicious access structures, where a subgroup of privileged shareholders in smaller number than the threshold can reconstruct the secret. We call this a hidden access structures (HAS). Let us supose that the Dealer choose a polynomial of degree 6 and generate from it an access structure. As noted in the example of Figure 2, there is a sub-access structure whose shares can interpolate the polynomial of degree 3 and whose independent term is the same than. In this case with less shares correctly assembled it is possible recontruct the secret.

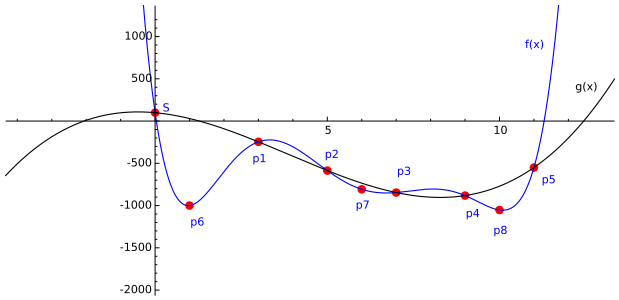


Figure 2. HAS

HAS can be created involuntarily in the moment of generation of access structure or the Dealer may be malicious and do it to take advantage in the future. In both cases we need to avoid a HAS, because it contradicts the fact that sets with cardinality smaller than t are not able to obtain information about the secret. We suggest to test the chosen polynomial and access structure generated by it.

Definition Hidden access structures (HAS) are subsets of the access structure with cardinality lower than t and that with it is possible to reconstruct the secret with it.

B. Hierarchy

It is possible to get a hierarchical scheme using tuples of shares [1]. For example, if we give the company president three shares, each vice-president two shares, and each executive one share, then a $(3, n)$-threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone. We call this a weighted threshold secret sharing scheme.

As we have seen, hidden access structures must be in general avoided. On the other hand, in some situations to have a sub-access structure such that can be useful. As we can see in Section III-A, one of the consequences that to work with the redundant share is the possibility of having more than one curve with the same independent term. This means that shares from the curve of degree less can be obtain more important status in the secret sharing scheme. We can to compare this situation with military hierarchy. For example, the shares from the curve of degree less can be generals and another shares are soldiers. Important to say that the generals can reconstruct the secret only if work together.

As example Figure 2 shows two polynomials, one of degree 6 and another of degree 3, both with the same independent term S. The shares p_1, p_2, p_3, p_4, p_5 are generals, so only with those shares it is already possible to reconstruct the secret. Thus, such shares should be considered hierarchically more important than the other shares.

Although, the Shamir's weighted threshold secret sharing scheme is more dynamic and flexible, our scheme gives this extra possibility while being simple already to apply.

 ISBN: 978-1-61208-427-5

C. Multi-secret

Although the use of a redundant share to determine the threshold implies the possibility of polynomial behave as false positive, since under the Dealer control, we can use this fact in our favor. In some situations more than one secret it is necessary.

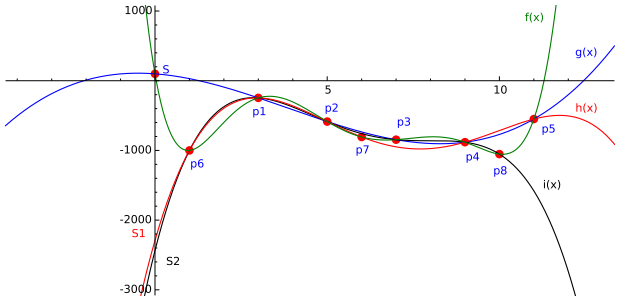


Figure 3. All curves

As we can see in Figure 3, we have 4 different possible polynomials, 2 with the same independent term and 2 with differents independent terms.

IV. Evaluation

According to theorem 5.1 of [9], computing an interpolating polynomial at shares can be performed with $O(n^2)$ operations in $\mathbb{F}[x]$. When using the Lagrange interpolation, for example, it is necessary to compute precisely $7n^2 - 7n$ operations. The Algorithm 1 to perform interpolation polynomial. In general, when only one round of interpolation is necessary, Lagrange interpolation is used.

The Algorithm 2 can use more than once Algorithm II-A, because the number of shares taken from the access structure is increased. It is possible to perform the interpolation polynomial in the Newton form and use the method of divided differences to construct the coefficients. One example is Neville's algorithm. The cost also is $O(n^2)$ operations. However, you only need to do $O(n)$ extra work if an extra share is added to the data set, while for the other methods, like Lagrange for example, you have to redo the whole computation.

It is easy to show that cost of the Algorithm 3 is $O(n^2 \times n!)$. The cost of the Algorithm 3 is significant hight to several shares. Thus, the next step of this work is to find more efficient algorithm. In spite of the cost be hight, using it we avoid the attacker to learn how much shares is necessary to bribe to break the system.

Although the proposed method is different from the traditional method, the security of the scheme is still based on the polynomial interpolation. In other words, with less than t shares, nobody can infer anything about the secret.

V. Conclusions and Future Works

One of the most significant findings to emerge from this study, is a method to determine the threshold value. In this way, the values of t and n do not need to be public information. This study has shown also a brief analysis of the possibility of construction an access structure that has different subsets to rebuild the secret. In other words, different subsets of shareholders will have different thresholds to rebuild the secret. Furthermore, in our investigations we have seen that is possible that there malicious access structures (HAS) where a previleged subgroup of shareholders in smaller numbers than the threshold can reconstruct the secret. In reliable secret sharing schemes HAS must be avoided. Thereby, our last propose is a method to do it.

Some intersting possible future works are presented next:

- In Section II we present a method to generate an access structure whose shares do not form false positive polynomials We belive thar is a more efficient method can be propose;
- To calculate the possibility of the false positive polynomials occurrence.

References

[1] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, Nov. 1979, pp. 612–613. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[2] G. R. Blakley, "Safeguarding cryptographic keys," Managing Requirements Knowledge, International Workshop on, vol. 0, 1979, p. 313.

[3] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 72, no. 9, 1989, pp. 56–64.

[4] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," in Advances in Cryptology CRYPTO 88. Springer, 1990, pp. 27–35.

[5] M. Mignotte, "How to share a secret," in Cryptography. Springer, 1983, pp. 371–375.

[6] A. Beimel, "Secret-sharing schemes: a survey," in Proceedings of the Third international conference on Coding and cryptology, ser. IWCC'11, 2011, pp. 11–46. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017916.2017918

[7] K. Wang, X. Zou, and Y. Sui, "A multiple secret sharing scheme based on matrix projection," in Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, vol. 1. IEEE, 2009, pp. 400–405.

[8] S. Iftene, "Secret Sharing Schemes with Applications in Security Protocols," Ph.D. dissertation, University of Iasi, Romenia, 2006.

[9] J. Von Zur Gathen and J. Gerhard, Modern computer algebra. Cambridge university press, 2013.

 ISBN: 978-1-61208-427-5

Enterprise Security Metrics with the ADVISE Meta Model Formalism

Brett Feddersen, Ken Keefe, William H. Sanders
Information Trust Institute
University of Illinois
Urbana, Illinois, USA
{bfeddrsn, kjkeefe, whs}@illinois.edu

Carol Muehrcke, Donald Parks
Cyber Defense Agency
Wisconsin Rapids, Wisconsin, USA
{cmuehrcke, cparks}@cyberdefenseagency.com

Andrew Crapo, Alfredo Gabaldon, Ravi Palla
General Electric Global Research
Niskayuna, New York, USA
{crapo, alfredo.gabaldon, palla}@ge.com

***Abstract*—Building secure, complex systems is a daunting task. The ADversary VIew Security Evaluation (ADVISE) formalism was designed to offer a model of an adversary attacking a system. As currently implemented in Möbius, ADVISE provides a rich and flexible system security model that, with the other features of Möbius, offers quantitative security metrics. For large systems, constructing realistic ADVISE models can be tedious and impractical. To remedy this issue, we propose the ADVISE meta modeling formalism. An ADVISE meta model is used, with the Möbius framework, to generate ADVISE models and other Möbius components from a higher level model constructed from components, adversaries, and metrics provided by associated Web Ontology Language libraries. This paper briefly reviews Möbius and ADVISE, then introduces the ADVISE meta modeling formalism.**

Keywords - ***Quantitative Security Analysis; State-based Security Model; Discrete Event Simulation; Adversary Behavior Model***

I. INTRODUCTION

Enterprise security metrics are a key component of any system design analysis. For over a decade, the Möbius framework has offered quantitative system performance and reliability metrics on Möbius models defined with formalisms such as Stochastic Activity Networks (SANs) and fault trees. Recently, the ADversary VIew Security Evaluation (ADVISE) formalism was added to the Möbius framework to provide a way to model attacks on a system by a variety of adversaries. With the ADVISE formalism and Möbius, enterprise-level, quantitative, security metrics can be measured on models of existing systems or systems still being designed.

While ADVISE has proven to be a useful approach, building large models can be difficult. Furthermore, when a system changes, reflecting those changes in the representative ADVISE model can be very time intensive. To alleviate this problem, we propose the ADVISE meta modeling formalism. An ADVISE meta model contains a higher level system diagram composed of component objects connected by relationship arcs. An ADVISE meta model also contains a set of adversaries, a set of security metrics, and a set of experimental configurations. We leverage the power of Web Ontology Language ontology descriptions to provide components, relationships, adversaries, and metrics to be used in ADVISE meta models.

The remainder of this extended abstract explains our approach and how we intend to validate it. In Section II, the Möbius framework is introduced. In Section III, we provide an overview of the current ADVISE modeling formalism. The new ADVISE Meta modeling formalism is explained is Section IV. We conclude with information about an Alpha trial we will be conducting in the near future to validate our approach in Section V.

II. THE MÖBIUS FRAMEWORK

The Möbius framework is a mature, extensible modeling and solution framework for discrete event systems. The Möbius tool combines the modeling formalisms and solution methods currently defined in the Möbius framework to offer a user-friendly graphical interface for defining complex system models, useful measures of the system, and a set of experiments. With the Möbius tool, these components are used by analytical solution techniques or the discrete event simulator to find values for the defined metrics.

In the Möbius framework, *atomic models* define the smallest pieces of the system being modeled. Several atomic models, using a mixture of modeling formalisms, can be defined to handle the necessary components of the system. Using one of the *composed modeling* formalisms implemented in Möbius, the defined atomic models, or several instances of an atomic model, can be joined together to build a complete, executable system model. With the *performance variables* formalism, various metrics based on time or system events can be created to quantitatively measure the behavior of the system model. A set of experiments are defined in a *study* to study the impact initial model parameters have on the behavior of the system. Finally, one of the analytical solvers can be used for certain classes of models or the discrete-event simulator for all models to generate results for the defined metrics.

III. THE ADVISE FORMALISM

The ADVISE atomic model formalism is composed of two parts: the attack execution graph and the adversary profile. These parts are necessary to defined the executable behavior of an adversary attacking a system and the effect those attacks have.

A. Attack Execution Graph

An attack execution graph details the attack surface of the modeled system. The graph consists of a set of attack steps, which are atomic actions that an adversary can choose to attempt. Upon completion of an attack step attempt, one of a set of outcomes defined on each attack step is stochastically selected and updates the model state. Model state is defined by the set of access, knowledge, skill, and goal elements defined in the attack execution graph. Each of the state elements represent whether or not the access or knowledge has been obtained, what degree of skill the adversary possesses, and whether or not a goal has been achieved.

 ISBN: 978-1-61208-427-5

B. Adversary Profile

The adversary profile of an ADVISE model details the initial model state, as well as the decision making ability and preferences of an adversary. A subset of access and knowledge elements are selected from the attack execution graph to indicate that the adversary possesses these at the beginning of the model's execution. A subset of skills, with a proficiency level for each skill, is defined to model how effective the adversary is at using these skills. A subset of the goals, with an associated payoff value for each, is also defined in the profile.

Part of the ADVISE method is the evaluation of the attack execution graph by the adversary in order to determine the attack path that will be attempted. This is done with a game-theoretic approach that evaluates the relative attractiveness of all possible attack paths to a defined depth. This depth is called that planning horizon and is an essential part of the adversary profile definition. A low planning horizon will result in a very fast model execution, but may limit the possible goals that can be achieved because the adversary may not explore deep enough to see the payoff from those deep goals. A large planning horizon can yield a slow model execution time.

In order for an adversary to evaluate the attractiveness of attack paths in the attack execution graph, three components of an attack path are considered: risk of detection, cost, and payoff. In the adversary profile, relative preference weights must be defined to model the adversary's interest in or aversion to one of those components. For example, a teenage hacker may not care about the costs of an attack the most and not very much about the risk of detection.

IV. The ADVISE Meta Formalism

The ADVISE formalism has been used to develop useful models of real systems and is currently being used for system security research by several organizations. However, the approach is not without its drawbacks. Attack execution graphs and defined manually by the modeler and can be quite time consuming for realistic definitions of large systems. Moreover, making changes to complex ADVISE models are also tedious. Also, past users frequently express uncertainty about their modeling decisions in various adversary profile parameters or whether or not their attack execution graph is complete enough to cover all potential attacks on a system.

To address these issues and more, we propose the ADVISE meta modeling formalism. The ADVISE meta model is a higher level model that can be used to generate ADVISE models. The ADVISE meta model consists of a *system diagram*, *adversary profile set*, *metric set*, and a set of *configurations*. From these components, one or more ADVISE models, performance variables models, studies, and discrete event simulators are generated in a Möbius project.

A. System Diagram

The system diagram is a graphical representation of the system being studied. The graph consists of blocks on a canvas that represent components of the system. The blocks are connected by relationship arcs. For example, if a server room is being modeled, an uninterruptable power supply and server may be blocks in the diagram and may be connected by a *poweredBy* relationship. Each component also has a set of attributes that can be defined on the component instance. For example, a server component may have an attribute that defines the number of processors possessed by the server. An ADVISE meta model may contain multiple system diagrams in order to easily study several structural variations of the same system.

B. Adversary Profiles

A set of adversary profiles are defined in a similar way to ADVISE models. These adversary profiles are later paired with a system diagram in the set of configurations. The key difference in an ADVISE meta model is that adversary profiles are no longer defined from scratch, but are instead selected from a structured library of adversaries included with the tool. Once an adversary template is selected, the profile is added to the adversary set and the user can make changes to the profile instance in the meta model.

C. Metrics

Similar to the adversary profile set, system security metrics do not need to be defined from scratch, but are rather selected from a library of metrics. Each metric has required attributed that must be defined by the user. For example, a metric that studies the average time until a specific server is compromised will need to choose a server instance from the system diagram to investigate.

D. Configurations

The final step in defining an ADVISE meta model is the set of configurations. A configuration matches a system diagram, an adversary profile, and a subset of the set of metrics. Each configuration will be used to generate an ADVISE atomic model and a performance variables model (with a performance variable for each of the metrics selected in the configuration).

V. Alpha Trial

As part of our ongoing research, we will be conducting an alpha trial of the ADVISE meta modeling formalism in late September, 2015. We are actively seeking interested participants from academia, government, and industry. Alpha participants will be given access to an alpha version of Möbius with the ADVISE meta formalism, as well as useful documentation for learning the tool. We will conduct regular conference phone calls with participants and work directly with organizations to make their experience a smooth one. We will provide a community exchange with a mailing list and wiki. We are hoping to receive useful feedback from participants to improve on the tool and method.

If you are interested in participating in the alpha trial, please contact us at advise@mobius.illinois.edu.

Acknowledgment

The work described here was performed, in part, with funding from the Department of Homeland Security under contract HSHQDC-13-C-B0014, "Practical Metrics for Enterprise Security Engineering."

 ISBN: 978-1-61208-427-5

Implementation Issues in the Construction of an Application Framework for Secure SMS Messages on Android Smartphones

Alexandre Melo Braga[12], Romulo Zanco Neto[1], André Luiz Vannucci[1], and Ricardo Shiguemi Hiramatsu[1]
[1] Centro de Pesquisa e Desenvolvimento em Telecomunicações (Fundação CPqD)
Campinas, São Paulo, Brazil
[2] Universidade Estadual de Campinas (UNICAMP)
Campinas, São Paulo, Brazil
Email:{ambraga,romulozn,vannucci,ricardoh}@cpqd.com.br

***Abstract*—This paper details the construction of an application framework for SMS security that provides secrecy, integrity, authentication, and non-repudiation for SMS messages. The proposed framework integrates authenticated encryption and short digital signatures to management services for cryptographic keys and digital certificates. The framework hides from final users all details concerning certificate and key management. A flexible trade-off between security objectives and message length makes it possible to offer three levels of security: (i) secrecy only, (ii) secrecy and message authentication, and (iii) secrecy, origin authentication and non-repudiation. The main contribution is the use of short signatures for SMS origin authentication, which makes it possible to pack in a single, 140-byte SMS all information necessary to authenticate the origin of encrypted messages, while the user is still left with a useful length of text.**

Keywords - SMS; Cryptography; Android; Security.

I. INTRODUCTION

Nowadays, despite the growing popularity of new message services, such as instant messages, on mobile devices, the old-fashion Short Message Service (SMS) is still in plenty of use. Due to its higher reachability, relatively low cost, small amount of traffic, and existing infrastructure, varied flavors of SMS applications are being used in various fields, such as mobile commerce [12], home automation [24], and Machine-to-Machine (M2M) communication [30]. Even though SMS is not suitable for real-time remote control, as it suffers from transmission delay, message lost and lack of confidentiality [24], the ordinary SMS technology has evolved from a simple messaging service to an integral component of these security-critical applications. The SMS persistent popularity can also be attributed to modern smartphone platforms, such as Google Android [20], that provide to software developers an easy means to include SMS functionality into mobile applications.

This paper details the construction of an application framework for SMS security that provides secrecy, integrity, authentication, and non-repudiation for SMS messages. The proposed framework integrates authenticated encryption and short digital signatures to management services for cryptographic keys and Initialization Vectors (IVs). The main contribution of this paper is the use of short signatures for SMS origin authentication. Those signatures are only 33-bytes long, but provide the same security level of conventional signatures with at least twice its size. Such a small length saves space in message payload, so that an authenticated message can occupy only a single SMS.

The resulting application framework for SMS security is part of an integrated infrastructure for mobile security on mobile devices [5][6], that provides strong cryptography [3][4] to security-aware mobile applications [1][2].

The text is organized as follows. Section II offers background information about SMS internal workings. Section III provides related work on SMS security. Section IV details de design of the proposed solution. Section V contains a performance evaluation. Section VI discusses implementation issues of the prototype. Section VII concludes the text.

II. BACKGROUND

A. SMS workings

The Short Message Service (SMS) is a standardized facility defined as part of the Global System for Mobile Communications (GSM) series of standards [21] and the following description is based upon that documentation. SMS enables the transmission of up to 1120-bit (140 bytes or 160 7-bit characters) alphanumeric messages between mobile phones or external systems. The SMS service provides out-of-band delivery of messages, meaning that user can receive or transmit messages also when a voice call or data transfer is already in progress. The delivery and the integrity of an SMS are guaranteed by the network even in presence of temporary failures or unavailable stations.

Any message, sent via SMS, is not directly delivered to its destination, but it is stored into an SMS Center (SMSC) after passing through a Mobile Switching Center (MSC), which has the important role of message routing. If the destination device is unavailable or not connected to the GSM network, the messages are stored in the SMSC and delivered when the destination becomes available again, through another MSC. SMS has got a validity period, for which it will wait for the destination device to be available. After that time, the SMSC will delete the message. (The usual validity period is one day).

By enabling the delivery confirmation option, disabled by default, the sender of a SMS message can receive a

 ISBN: 978-1-61208-427-5

return message notifying whether it has been successfully delivered. Without the notification option, there is no guarantee about the correct reception or in the correct order of delivery of a previously sent SMS message.

Techniques for SMS concatenation and compression have been defined and incorporated in the GSM standard. However, these features may not be reliably implemented worldwide. Thus, only single uncompressed message delivery should work everywhere.

B. SMS on Android

Google's website on Android [20] is a well-documented source of information for software developers. According to that documentation [20], the basic building blocks of Android's internal messaging system are Intents. The Intent is a simple message object that holds additional information about an operation to be performed or of an event that has happened. Upon reception of an SMS message, intents are used to broadcast the contents of the received SMS message internally to registered receivers. To receive a broadcasted intent, an Android app needs to implement and register an appropriate broadcast receiver. The registered broadcast receiver's implementation determines the actions to be carried out, when a broadcast is received. Within an Android app, broadcast receivers can be registered either statically or dynamically.

If multiple broadcast receivers are registered for the same intent, they are called according to their given priorities. It is important to note that registered receivers have the possibility to abort a received broadcast intent, which prevents the intent from being forwarded to further registered receivers with a lower priority.

C. GSM encryption and SMS security

Global System for Mobile (GSM) communications has point-to-point encryption based upon the A5 family of ciphers [21]. A5/1 is the original cipher designed for use in the GSM protocol. After several weaknesses had been discovered, this cipher is not considered secure anymore [16]. A5/2 is a weakened version of A5/1 to allow for (historically kept) export restrictions to certain countries. It is therefore not considered secure [16]. A5/3 is also called Kasumi and is still in use today. Currently, known attacks do not inhibit its practical use in GSM. However, it is not intended to be used in further applications [16]. Though SMS encryption could be provided by network layer, on a point-to-point basis, it should be noted that end-to-end encryption of SMS still belongs to the application layer.

SMS technology is also inherently insecure, since the messages could be intercepted during the path that the messages follow during the transmission, including base stations, SMSC, and MSC. Particularly, SMS is subjected to replay attacks, message falsification, payload corruption, and sender impersonation.

Inside mobile devices, SMS sniffers and SMS catchers weaken the security of received and processed SMS messages, as these kinds of malware can simply use available APIs to intercept SMS messages [37]. SMS sniffers intercept SMS messages to spy on their content. After interception, the original message is forwarded to the default SMS-processing application (receiver). Differently, SMS catchers discard the message after its interception. This way, SMS catchers hide the original message from the user. This kind of malware intercepts incoming SMS messages either to spy on security-sensitive data transmitted via SMS or to receive SMS-based malware control commands [37]. The architecture of the Android platform facilitates the implementation of SMS catcher and sniffers even on non-rooted smartphones [37].

Starting with version 4.4, Android [20] enforces the forwarding of incoming SMS messages to a default SMS application. This renders realization of SMS catchers more difficult but by no means prevents SMS sniffers, as installed applications are still capable of reading incoming SMS messages [37].

III. RELATED WORK

The development of end-to-end encryption of SMS by mobile applications has a history of at least ten years and its evolution seems to coincide with the proliferation of mobile application platforms, as shown in next paragraphs.

In 2005, Hassinen presented SafeSMS [26] as an application meant for SMS end-to-end encryption. Encryption was based on a secret password shared between the sender and the recipient. The shared secret could also be generated and distributed by the system using a key exchange procedure, or by using a few text messages in a key exchange protocol, such as the Diffie-Hellman (DH).

In 2008, Lisonek and Drahansky [14] describe an application for securing of SMS which prevents tapping and substitution of SMS messages, by using RSA with OAEP padding. Also, Hossain et al [8] proposed a security scheme for improving the SMS security by encrypting SMS messages using GSM encryption technology, and digitally signing them with public key signature.

In 2009, Kuaté, Lo, and Bishop [36] proposed a supposed simple-to-use but cryptographically strong API for message encryption and authentication, called Linca, which was designed for limited devices, as well as a protocol called SMSSec for confidentiality and integrity.

In 2010, De Santis et al [7] presented Secure Extensible and Efficient SMS (SEESMS), a software framework written in Java, which allows two peers to exchange encrypted and digitally signed SMS messages. SEESMS supports the encryption of a communication channel through the Elliptic Curve Integrated Encryption Scheme (ECIES) and the Rivest-Shamir-Adleman (RSA) algorithms. The identity validation of the contacts involved in the communication is implemented through the RSA, Digital Signature Algorithm (DSA) and Elliptic Curve DSA (ECDSA) signature schemes. Also, Read and Martina presented SAMES [15], an Android application that allows

 ISBN: 978-1-61208-427-5

its users to send and receive text messages that are encrypted with the AES using hashing algorithms or Elliptic Curve DH (ECDH) for key creation. Certificates can also be created, signed, verified and sent to others using text messages. However, producing a method to convert certificates to a string less than 140 characters was not possible due to the signature itself exceeding this limit. In addition, Agoyi and Seral [25] compared RSA, ELGamal and Elliptic Curve Cryptography (ECC). They concluded, at the time (2010), that large key size algorithms were not suitable for SMS encryption due to small memory and low computational power of mobile phones. This has put ECC at an advantage over RSA and ELGamal in SMS encryption.

In 2011, Nanda and Awasthi analyzed Joint Channel Coding and Cryptography (JCCC) and Soft Input Decryption (SID) and proposed two algorithms to be used in SMS security: NTRUSign [9] and XTR – NR Signature [10]. Also, Qi, Pan, and Ding [31] used FPGA to implement the RSA algorithms and applied it in SMS encryption system. The implementation not only encrypted the SMS with hardware encryption, but also used a public-key server.

In 2012, Saxena and Chaudhari performed research [33] in securing SMS with a variant of ECDSA. Also, Saxena, Chaudhari, and Prajapati [34] proposed an encryption approach that used a password-based key exchange protocol based on DH and generated a shared secret-key which could be used in message encryption as well as in MAC functions.

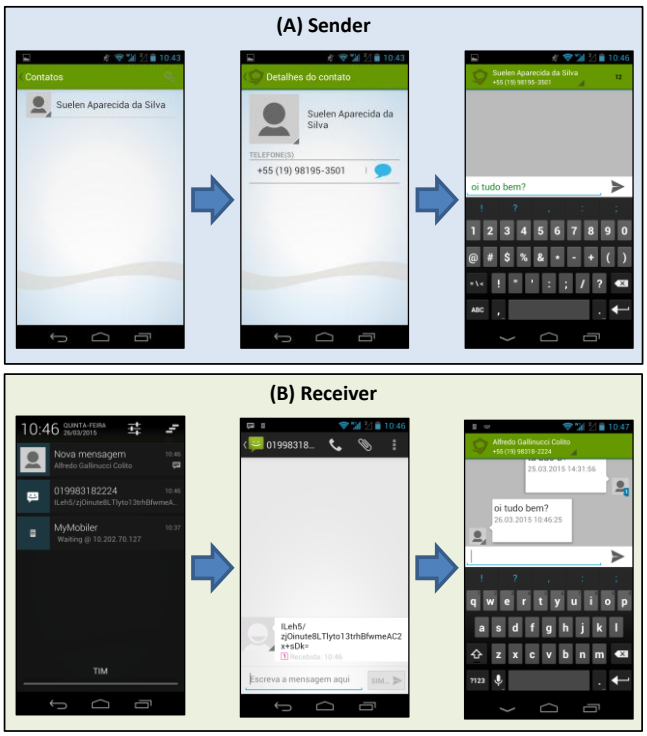


Figure 1. Screenshots showing the steps for sending and receiving encrypted SMS messages. In (A), Sender finds a contact, selects her, writes a message and sends it to her. In (B), Receiver is notified about an incoming message from a known contact, the message is shown encrypted by Android and can be decrypted only by CryptoSMS app.

Still in 2012, Chavan and Sabnees [39] proposed a technique that combines encryption and compression. The technique encrypts the SMS using ECC and after that, the encrypted SMS is compressed using a lossless algorithm. The supposed advantage is the decreasing of message lengths while maintaining the security. In addition, Pan, Ding, and Qi, [23] proposed a chaos-based encryption scheme combined with A5/1 algorithm for SMS security. The solution was tested on a phone-like system designed in Field-Programmable Gate Array (FPGA).

In 2013, Pereira et al [19] implemented SMSCrypto, a Java framework for securing SMS-based communications in mobile phones. The framework encloses a tailored selection of lightweight cryptographic algorithms and protocols, providing encryption, authentication, and signature services. Also, Khan, Bakhtiari, and Bakhtiari [28] proposed a framework that uses HTTPS for secure key exchange, as well as ECC and RSA as encryption algorithm to protect SMS messages against MITM attacks.

Still in 2013, Ariffi, Mahmod, Rahmat, and Idris [40] dealt with SMS encryption on Android smartphones. They proposed the use of a block cipher called 3D-AES for SMS encryption. Sagheer, Abdulhameed, and AbdulJabbar [11] proposed a solution for confidentiality and integrity for SMS by applying a hybrid cryptographic scheme which combines Advanced Encryption Standard (AES) for encryption and Rivest Cipher 4 (RC4) for key expansion and generation. It was implemented in Java and tested on a Nokia 5233. Pizzolante et al [38] investigates the feasibility of secure file transfer through SMS. Their solution compresses and eventually encrypts a variable-sized massage (or file) and sends it through standard SMS.

Again, in 2013, Saxena, Chaudhari, and Thomas [35] provided solutions to the repudiation attack on SMSs by using a variant of ECDSA. In 2014, Saxena and Chaudhari [32] proposed a protocol called EasySMS which provides end-to-end security during the transmission of SMS. This solution puts key management on the control of Mobile Network Operator (MNO).

In 2014, three unusual papers have appeared. First, Fahrianto, Masruroh, and Ando [17] argued that a combination of two “toy” ciphers (Caesar and Vinegére) was good enough to protect the secrecy of SMS, which is hardly believable. Second, Kashif [27] rediscovers RSA encryption for SMSs. However, the paper gives no clue about key management, randomization of RSA, and message splitting. Third, Al Bashar Abul Ulayee, Mesbah-Ul-Awal, and Newaj [22] used Caesar cipher in CBC mode to encrypt SMSs, and used a MAC for message authentication. The paper does implement a proprietary key management, arguing that the method is sufficient to surpass the weakness of Caesar Cipher, which is an unlikely assumption.

Finally, still in 2014, Patil, Sahu, and Jain [29] studied SMS compression in order to minimize the overhead of

 ISBN: 978-1-61208-427-5

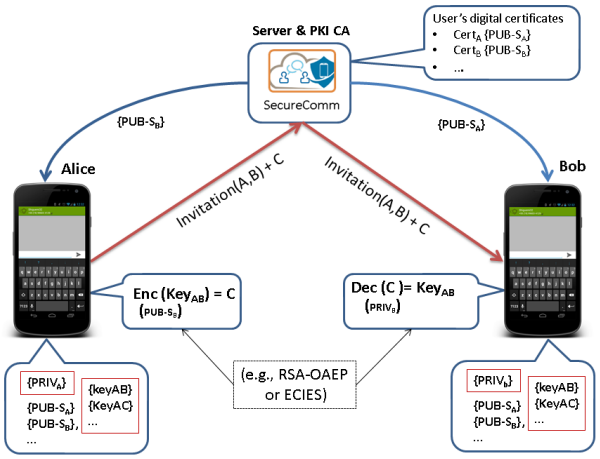


Figure 2. Shared-key transport is an invitation to a fellow to become a contact.

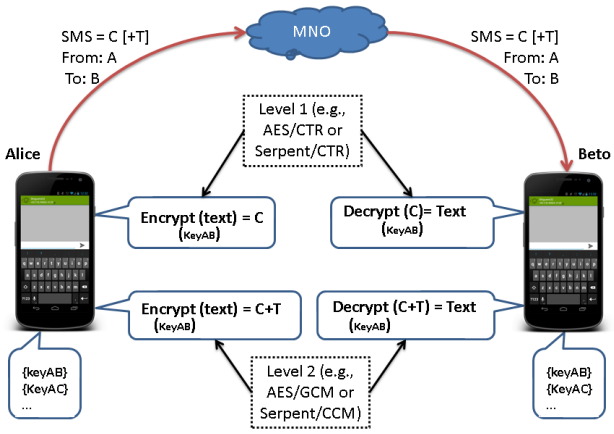


Figure 3. Levels 1 and 2 – encryption and authenticated encryption.

payload due to encryption, and proposed a method for compression of SMSs after encryption using ECC.

IV. PROPOSED SOLUTION

This section details the construction of the application framework for SMS security that provides encryption, integrity, authentication, and non-repudiation for SMS messages. From now, the framework is called CryptoSMS. CryptoSMS is not a stand-alone mobile application. In fact, it is supported by a server-side Java Enterprise Edition (JEE) application for management of users, apps and trust, which is integrated to a XMPP-based message service, a Public-Key Infrastructure (PKI) and Certification Authority (CA).

The general usage of CryptoSMS is quite simple, as seen in Figure 1, which shows screenshots for both sending and receiving of encrypted SMS messages. In Figure 1(A), Sender finds a contact, selects her, writes a message, and sends it to her. In Figure 1(B), the Receiver is notified about an incoming message from a known contact, the message is shown encrypted by Android (as was expected), and could only be decrypted by CryptoSMS.

Modern mobile platforms, such as Android, allow users to extend their device's behavior by installing new mobile apps. App installation is an import event, from service provider's point of view, and can be security sensitive. Not only app's binary code has to be accepted by the host device, but generally user accounts, along with other data, have to be created and registered as a profile for the new user.

CryptoSMS performs security sensitive actions during app installation, such as user account creation, key pair generation, public key upload, and (predefined) contacts synchronization, including the download of contact's digital certificates. Certification authority's root certificates are embedded in the app and distributed along with it.

It is a well-accepted behavior in social networks that users have contact lists (or friends) and that a pair of users becomes friends after an invitation is sent and accepted. In CryptoSMS, users' relationships behave like social networks contacts. This is an important usability aspect, as it resembles similar communication systems and utilizes the same software components, such as Android's notification service, to inform the user about new invitations.

The invitation process, illustrated in Figure 2, triggers several security related actions. For instance, when Alice sends to Bob an invitation to share secure SMSs, this request is supplemented by the generation of a secret key (K_{AB}), its encryption with Bob's public-key, and its secure transfer to CryptoSMS server. At this point, an asymmetric algorithm, such as RSA-OAEP or ECIES, can be used to securely transport K_{AB}. Bob's acceptance of Alice's invitation triggers the download of key K_{AB} to Bob's device.

The process for securing SMS messages depends on the desired security level and is illustrated in two parts by Figure 3 and Figure 4. The security level is not related to key sizes, as is usual in cryptography, since only 256-bit security is used for cryptographic algorithms. On the other hand, Security levels try to capture the security perception of the user and are related to the security objective of secrecy, integrity, authentication and non-repudiation, as follows:

- Level one grants only secrecy and was implemented with a symmetric block cipher (e.g., AES or Serpent) in in CTR mode, as shown in Figure 3;
- Level two grants secrecy, integrity, and message authentication, and was implemented with algorithms for symmetric authenticated encryption (e.g., AES/GCM or Serpent/CCM), as shown in Figure 3;
- Level three, illustrated in Figure 4, grants not only secrecy, integrity, and message authentication, but also grants user authentication and non-repudiation.

In level three, user authentication and non-repudiation of messages are accomplished by an unusual kind of digital signatures called short signatures, which are provided by asymmetric cryptographic algorithms and, as such, require management of key pairs and authenticated distribution of public keys. All cryptographic implementations are provided by a proprietary Cryptographic Service Provider (CSP) [3].

This security level three was implemented with a symmetric block cipher (e.g., AES or Serpent) in CTR mode and a short signature, such as Boneh-Lynn-Shacham (BLS) [13] or Zhang-Safavi-Susilo (ZSS) [18]. It is interesting to note that the short signatures used by CryptoSMS are only

 ISBN: 978-1-61208-427-5

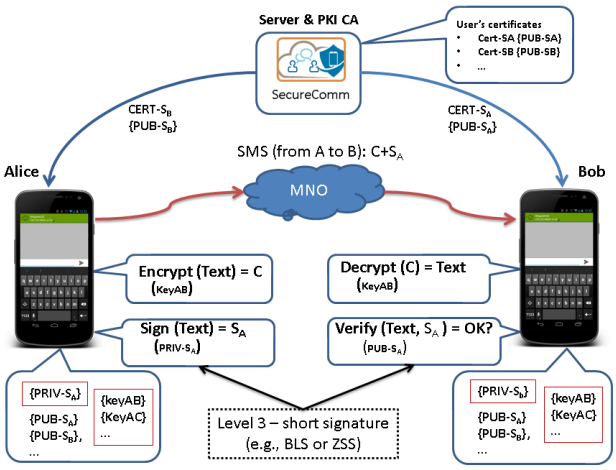


Figure 4. Security level 3 – encryption and short signatures.

33-byte long, but provide the same security of conventional, 66-byte ECDSA and are stronger then 256-byte RSA.

The resulting secure SMS carries with it an encrypted payload (in level one) and an authentication tag (in level two) or a short signature (in level three). Secret keys, public-key pairs, digital certificates, and other cryptographic parameters are all managed locally at the Android device by an app container fully integrated (synchronized) to CryptoSMS server and CA. It is an interesting usability aspect that users do not have to deal with cryptographic material, because it is all hidden from final user behind the concepts of invitations, contacts, data syncs, app installation, updates, and notifications.

V. PERFORMANCE EVALUATION

This section evaluates the performance in seconds taken to deliver an SMS message from one device to another. The performance of short signatures compared to conventional cryptography is also evaluated.

Figure 5(A) shows time measurements in seconds for SMS delivery from sender's device (dev1), through SMSC, to receiver's device (dev2). SMSC is the SMS routing center, a network entity that receives SMS messages from sender, temporarily stores them, and forwards them to receiver. Both sender and receiver were under the same SMSC, so network routing was minimal and latency was mostly due to network congestions.

A total of 20 SMS messages were sent in a loop of 50 iterations, resulting in 1,000 SMSs. There were no message losses or message disordering (rearrangement of message sequence during arrival). The overall time for SMS delivery stood around 20 seconds in average, and 90% of the values stood below that average. It is interesting to observe that the time taken from dev1 to SMSC is always balanced by the time taken from SMSC to dev2, preserving the average time.

Figure 5 also shows time measurements in milliseconds (ms) for two types of cryptographic services: symmetric encryption and digital signatures. The measurements were taken on a Samsung Galaxy S III (Quad-core 1.4 GHz Cortex-A9 processor, 1GB of RAM, and Android 4.1). Figure 5(B) shows time measurements of single-block encryption and decryption for Serpent and AES. Algorithms were setup with a 256-bit key. The bar chart shows the 9th centile of 10 thousand operations. Serpent is faster than AES.

Figure 5(C) shows generation of digital signatures for four algorithms: RSA (1024-bit key), ECDSA (with SHA-256), BLS, and ZSS, all of them for 256-bit security. For signature generation, RSA is the slowest one. Elliptic Curve Cryptography (ECC), as in ECDSA, is faster. Short signatures, such as BLS and ZSS, are not as fast as ECC. For signature verification, RSA is the fastest one, ECDSA is not that fast, and BLS/ZSS are the slowest ones.

Experiment has shown that network latency for SMS delivery (in seconds) is much larger than encryption and digital signature operations (in milliseconds). It is interesting to notice that the long latency for SMS delivery inhibits the use of key agreement protocols for negotiation of ephemeral keys through the SMS channel and it is an explanation for using pre-distributed secret keys.

BLS and ZSS are not as fast as ECC, since their constructions are based on complex mathematical structures

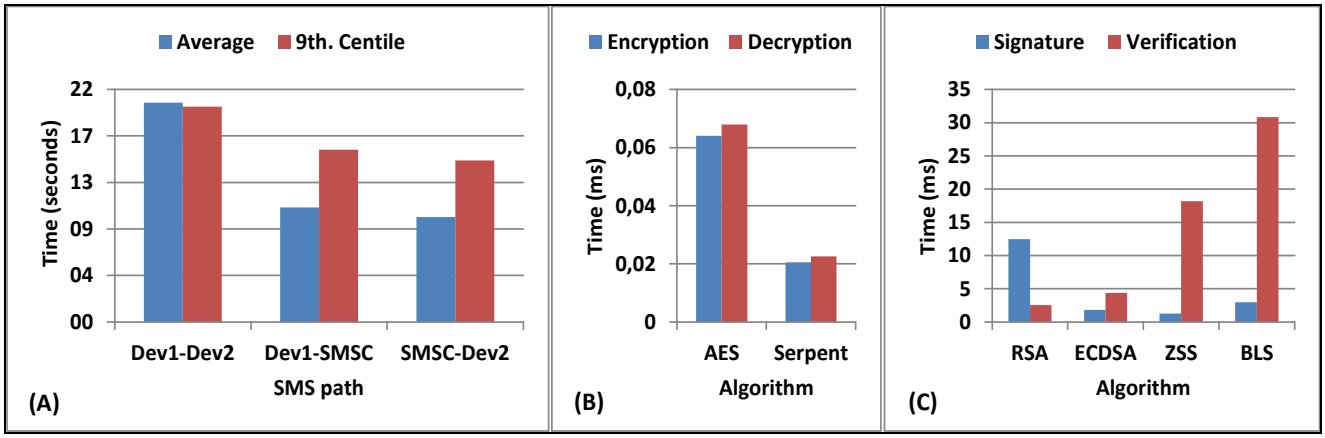


Figure 5. Three bar charts showing time measurements. In (A), time elapsed to transmit SMS messages from sender's device (dev1), through SMSC, to receiver's device (dev2). In (B), time for encryption and decryption of messages using AES and Serpent. In (C), time for digitally sign and verify signatures of messages with 64-byte length using RSA-PSS, ECDSA with SHA-256, ZSS, and BLS.

 ISBN: 978-1-61208-427-5

called bilinear pairings that require more computations. Here, there is a tradeoff, because the short signature can be roughly half the size of a regular ECDSA signature, but the verification is less efficient.

VI. DISCUSSION

This section discusses important implementation issues of CryptoSMS framework. One of them, the incorrect use of hardcoded Initialization Vectors (IVs), even with fixed or constant values, is a frequent issue on mobile devices. According to a NIST standard [41], the Counter (CTR) mode requires unique IVs and this constraint is inherited by authenticated encryption with Galois/Counter mode (GCM).

CryptoSMS implements IV management services in order to fulfill the uniqueness requirement for CTR, GCM and CCM modes of operation, as illustrated by Figure 6. During the invitation process, an IV initialization package is generated by Alice and saved at server until Bob retrieves it. The IV is split in two: a base and a count. The IV package consists of a base IV for Alice, a base IV for Bob, and a nonce to be used by both Alice and Bob as an initial counter.

Another aspect is the order in which encryption and authentication is performed over plaintext. Authenticated encryption (e.g., GCM and CCM) does not suffer from such an issue because encryption and message authentication are embedded in a single service. However, short signatures have to be programmatically composed with encryption by the application programmer. The correctly way to make this composition is to encrypt the message, then sign the encrypted message. This method provides not only integrity of cipher text and plaintext, but also does not provide any side information on the plaintext.

Base64 encoding is used as a sanitization measure in order to avoid misinterpretation of binary SMS messages by other receiver apps running at the same device, besides CryptoSMS. This measure reduces the total length of a single message from 140 to 105 bytes. After excluding the length of a 20-byte tag, the user is left with 85 bytes of text length. When a 33-byte signature is used instead, the user has 72 bytes of text length.

CryptoSMS makes it possible to pack in a single, 140-byte SMS message an encrypted payload along with its authentication tag or short signature. The payload could be at most 105-bytes long, which is enough for a large number of applications. Split the text in a sequence of SMSs is a way to circumvent this limitation. However, transmission delay, message lost, and reception out of order of the message sequence, may cause incorrect decryption.

Finally, a last concern is that CA software had to be customized to support digital certificates for non-standard algorithms. This means that certificates has to be generated and verified for public-keys of short-signature algorithms.

VII. CONCLUDING REMARKS

This paper discussed the construction of an application framework for SMS security on Android smartphones. The framework provides secrecy, integrity, authentication, and non-repudiation for SMS messages. The use of authenticated encryption and short digital signatures makes it possible to pack in a single SMS all necessary information to authenticate encrypted SMSs, while preserving a useful length of messages. The usability of security features is addressed by offering easy-to-use security levels and a seamless integration of cryptographic management into common app management functions.

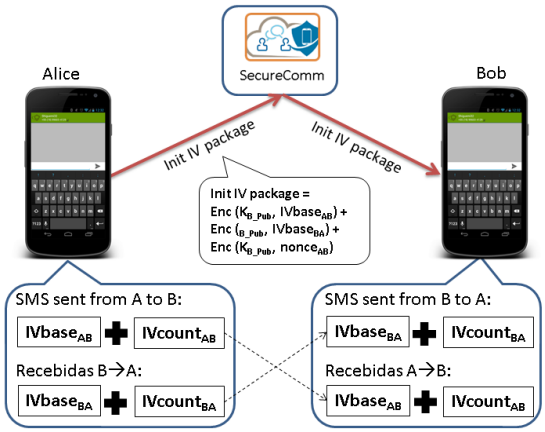


Figure 6. IV management for SMS security.

ACKNOWLEDGMENT

The authors acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11.0028.00 with the Financier of Studies and Projects - FINEP / MCTI.

REFERENCES

[1] A. Braga and A. Colito, "Adding Secure Deletion to an Encrypted File System on Android Smartphones," in proc. of The 8th Int'l Conf. on Emerging Security Inform., Systems and Technologies (SECURWARE), 2014, pp. 106–110.

[2] A. Braga and D. Schwab, "Design Issues in the Construction of a Cryptographically Secure Instant Message Service for Android Smartphones," in proc. of The 8th Int'l Conf. on Emerging Security Information, Systems and Technologies (SECURWARE), 2014, pp. 7–13.

[3] A. Braga and E. Morais, "Implementation Issues in the Construction of Standard and Non-Standard Cryptography on Android Devices," in proc. of The 8th Int'l Conf. on Emerging Security Information, Systems and Technologies (SECURWARE), 2014, pp. 144–150.

[4] A. Braga and E. Nascimento, "Portability evaluation of cryptographic libraries on android smartphones," in proc. of Cyberspace Safety and Security (CSS), 2012, pp. 459–469.

[5] A. Braga, "Integrated Technologies for Communication Security on Mobile Devices," in proc. of the 3rd Int'l Conf. on Mobile Services, Resources, and Users (Mobility), 2013, pp. 47–51.

[6] A. Braga, E. Nascimento, and L. Palma, "Presenting the Brazilian Project TSAM–Security Technologies for Mobile Environments," in proc. of Security and Privacy in Mobile Information and Comm. Systems, no. i, 2012, pp. 53–54.

[7] A. De Santis, A. Castiglione, G. Cattaneo, M. Cembalo, F. Petagna, and U. F. Petrillo, "An Extensible Framework for Efficient Secure SMS," in proc. of The Int'l Conf. on

 ISBN: 978-1-61208-427-5

Complex, Intelligent and Software Intensive Systems (CISIS), 2010, pp. 843–850.

[8] A. Hossain, S. Jahan, M. M. Hussain, M. R. Amin, and S. H. S. Newaz, "A proposal for enhancing the security system of short message service in GSM," in proc. of The 2nd Int'l Conf. on Anti-counterfeiting, Security and Identification (ASID), 2008, pp. 235–240.

[9] A. K. Nanda and L. K. Awasthi, "Encryption based channel coding algorithm for secure SMS," in proc. of The World Congress on Information and Communication Technologies (WICT), 2011, pp. 1282–1287.

[10] A. K. Nanda and L. K. Awasthi, "Joint Channel Coding and Cryptography for SMS," in proc. of The Int'l Siberian Conf. on Control and Communications (SIBCON), 2011, pp. 51–55.

[11] A. M. Sagheer, A. A. Abdulhameed, and M. A. AbdulJabbar, "SMS Security for Smartphone," in proc. of The 6th Int'l Conf. on Developments in eSystems Engineering (DeSE), 2013, pp. 281–285.

[12] A. Pourali, "The presentation of an ideal safe SMS based model in mobile electronic commerce using encryption hybrid algorithms AES and ECC," in proc. of The 8th Int'l Conf. on e-Commerce in Developing Countries: With Focus on e-Trust (ECDC), 2014, pp. 1–10.

[13] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J.Cryptol., vol. 17, n. 4, 2004, pp.297–319.

[14] D. Lisonek and M. Drahansky, "SMS Encryption for Mobile Communication," in proc. of The Int'l Conf. on Security Technology (SECTECH'08), 2008, pp. 198–201.

[15] D. Read and J. Martina, "SAMES-Short Anonymous Message Encryption Scheme," in proc. of X Simpósio Brasileiro em Segurança da Informação e de Sistemas computacionais (SBSeg), Fortaleza, Ceará, Brasil, 2010.

[16] ENISA, "Algorithms, key size and parameters report", nov. 2014. Retrived [July 2015] from www.enisa.europa.eu/ activities/identity-and-trust/library/deliverables/algorithms-key-size-and-parameters-report-2014.

[17] F. Fahrianto, S. Masruroh, and N. Ando, "Encrypted SMS application on Android with combination of caesar cipher and vigenere algorithm," in proc. of The Int'l Conf. on Cyber and IT Service Management (CITSM), 2014, pp. 31–33.

[18] F. Zhang, R. Safavi-Naini, and W. Susilo, "An Efficient Signature Scheme from Bilinear Pairings and Its Applications," in F. Bao, R. H. Deng and J. Zhou, ed., 'Public Key Cryptography', 2004, pp. 277-290.

[19] G. C. C. F. Pereira et al., "SMSCrypto: A lightweight cryptographic framework for secure SMS transmission," in Jour. of Sys. and Software, vol. 86, no. 3, 2013, pp. 698–706.

[20] Google, Inc., "The Android Project," Retrived [July 2015] from http://www.android.com.

[21] GSM Doc 28/85 "Services and Facilities to be provided in the GSM System" rev2, June 1985.

[22] H. Al Bashar Abul Ulayee, M. Mesbah-Ul-Awal, and S. Newaj, "Simplified Approach Towards Securing Privacy and Confidentiality of Mobile Short Messages," in proc. of The 4th Int'l Conf. on Advanced Computing Communication Technologies (ACCT), 2014, pp. 403–408.

[23] J. Pan, Q. Ding, and N. Qi, "The Research of Chaos-based SMS Encryption in Mobile Phone," in proceedings of The 2nd Int'l Conf. on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012, pp. 501–504.

[24] L. Pu, "An Improved Short Message Security Protocol for Home Network," in proc. of The Int'l Conf. on Future Computer and Communication, (FCC'09), 2009, pp. 62–65.

[25] M. Agoyi and D. Seral, "SMS Security: An Asymmetric Encryption Approach," in proceedings of The 6th Int'l Conf. on Wireless and Mobile Communications (ICWMC), 2010, pp. 448–452.

[26] M. Hassinen, "SafeSMS - end-to-end encryption for SMS," in proc. of the 8th Int'l Conf. on Telecommunications, (ConTEL), vol. 2, 2005, pp. 359–365.

[27] M. Kashif, "Secure SMS Communication Using Encryption Gateway and Digital Signature," in proc. of The 17th IEEE Int'l Conf. on Computational Science and Engineering (CSE), 2014, pp. 1430–1434.

[28] M. M. Khan, M. Bakhtiari, and S. Bakhtiari, "An HTTPS approach to resist man in the middle attack in secure SMS using ECC and RSA," in proc. of The 13th Int'l Conf. on Intelligent Sys. Design and Appl. (ISDA), 2013, pp. 115–120.

[29] M. Patil, V. Sahu, and A. Jain, "SMS text Compression and Encryption on Android O.S," in proc. of The Int'l Conf. on Computer Comm. and Informatics (ICCCI), 2014, pp. 1–6.

[30] N. Gligoric, T. Dimcic, D. Drajic, S. Krco, and N. Chu, "Application-layer security mechanism for M2M communication over SMS," in proc. of The 20th. Telecommunications Forum (TELFOR), 2012, pp. 5–8.

[31] N. Qi, J. Pan, and Q. Ding, "The Implementation of FPGA-based RSA Public-key Algorithm and its Application in Mobile-phone SMS Encryption System," in proc of The 1st Int'l Conf. on Instrumentation, Measurement, Computer, Communication and Control, 2011, pp. 700–703.

[32] N. Saxena and N. S. Chaudhari, "EasySMS: A Protocol for End-to-End Secure Transmission of SMS," in proc. of The IEEE Transactions on Information Forensics and Security, vol. 9, no. 7, 2014, pp. 1157–1168.

[33] N. Saxena and N. S. Chaudhari, "Secure encryption with digital signature approach for Short Message Service," in proc. of The World Congress on Information and Communication Technologies (WICT), 2012, pp. 803–806.

[34] N. Saxena, N. S. Chaudhari, and G. L. Prajapati, "An extended approach for SMS security using authentication functions," in proc. of The 7th IEEE Conf. on Industrial Electronics and Applications (ICIEA), 2012, pp. 663–668.

[35] N. Saxena, N. S. Chaudhari, and J. Thomas, "Solution to an attack on digital signature in SMS security," in proc. of The 5th Int'l Conf. on Modeling, Simulation and Applied Optimization (ICMSAO), 2013, pp. 1–6.

[36] P. H. Kuaté, J. L.-C. Lo, and J. Bishop, "Secure asynchronous communication for mobile devices," in proc. of the Warm Up Workshop for ACM/IEEE ICSE 2010, 2009, pp. 5–8.

[37] P. Teufl, T. Zefferer, C. Woergoetter, A. Oprisnik, and D. Hein, "Android - On-device detection of SMS catchers and sniffers," in proc. of The Int'l Conf. on Privacy and Security in Mobile Systems (PRISMS), 2014, pp. 1–8.

[38] R. Pizzolante, B. Carpentieri, A. Castiglione, A. Castiglione, and F. Palmieri, "Text Compression and Encryption through Smart Devices for Mobile Communication," in proc. of The 7th Int'l Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013, pp. 672–677.

[39] R. R. Chavan and M. Sabnees, "Secured mobile messaging," in proc. of The Int'l Conf. on Computing, Electronics and Electrical Technologies (ICCEET), 2012, pp. 1036–1043.

[40] S. Ariffi, R. Mahmod, R. Rahmat, and N. A. Idris, "SMS Encryption Using 3D-AES Block Cipher on Android Message Application," in proc. of The Int'l Conf. on Advanced Comp. Science App. and Tech. (ACSAT), 2013, pp. 310–314.

[41] NIST SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. 2007. Retrived [July 2015] from http://csrc.nist.gov/publications/ nistpubs/800-38D/SP-800-38D.pdf.

 ISBN: 978-1-61208-427-5

A Review and Analysis on Heartbleed on Italian Websites, a Year Later

Vito Santarcangelo[1,2] Fabrizio Valenti[2] Muhammad Imran Tariq[3] Claudio Fornaro[4]
Giuseppe Oddo[1,2]
Domenico Di Carlo Jr.[1]

[1]Centro Studi S.r.l.
Zona Industriale
Buccino, Italia

[2]iInformatica S.r.l.s.
Corso Italia,77
Trapani, Italia

[3]Dept. of Info. Technology
Superior University Lahore
Lahore, Pakistan

[4]Universitá Telematica Uninettuno
Corso V. Emanuele II
Roma

***Abstract*—Heartbleed, a big Open Secure Socket Layer (OpenSSL) vulnerability appeared on the web on 7th April 2014. This highly risked vulnerability enabled attackers to remotely read protected memory contents from Hyper Text Transfer Protocol Secure (HTTPS) sites. In this paper, the authors will review and analyze Heartbleed vulnerability effects on secured websites, a year later (April 2015). To accomplish this, we conducted an analysis on a dataset of 100 Italian public and private sector websites like banks, stock exchanges, Cloud Organizations and services on HTTPS websites, thereby obtained that only 1% of the websites show the vulnerability. However, new vulnerabilities as Padding Oracle on Downgraded Legacy Encryption (POODLE) & Factoring Attack on RSA-Export Keys (FREAK) affect a lot of websites, particularly the websites used as point of accesses of Italian telematics process. We concluded the paper with the analysis of the Cloud risks that are very harmful for the Cloud customers as well as the Cloud venders due to Heartbleed attack.**

Keywords–Heartbleed; OpenSSL; Poodle; Freak; Vulnerability.

I. INTRODUCTION

The Cyber Security is living an awkward moment caused by transition from the technical community to the public one [3]. Cyber Security is one of the most important topic in our days, because people which interact with the external world have to navigate in a secure mode. Heartbleed is a famous security bug spread in April 2014 [6]. It has caused a serious vulnerability bug in an open source cryptographic software library called OpenSSL. This library implements Transport Layer Security (TLS), an upgrade of Secure Socket Layer (SSL), a protocol developed to guarantee and to provide communication security between two or more devices over the Transmission Control Protocol (TCP) networks. This bug has involved the commercial transactions on the Internet of hundred million people around the world. It allowed everyone to listen to secure traffic exchanged between endusers. In this way, attackers steal the traffic (such as access credentials, passwords, payment cards of the users) by using secret keys used for traffic encryption [1]. It is considered one of the devastating disasters occurred in the internet age. This bug is known as Heartbleed because it uses the Transport Layer Security (TLS) protocol heartbeat extension; when it is broken, the secure communication channel between server and client is altered giving attackers the possibility to gather data. Heartbleed is the most famous cyber-attack in the last years.

A. Transport Layer Security (TLS)

The Transport Layer Security (TLS) is a protocol specified by Internet Engineering Task Force (IETF) as an enhancement over Netscapes Secure Socket Layer in 1999 [4]. It manages encryption and authentication on the TCP networks. Its peculiarity is enforcing security and data integrity. TLS protocol uses some kind of algorithms called ciphers to guarantee data integrity. TLS, in order to create a secure channel communication, uses a Public Key Infrastructure (PKI). As a consequence, the remote peers exchange information using an asymmetric cryptography mode. Each peer has two different keys: a private key and a public key. The first one is used by senders to sign digitally messages through a digest. The second one is used by receivers to validate the messages integrity. They create a new digest and then they compare it with the digest received message of the public key. If the two digests match, the message will be validated and verified. Two end peers can exchange information along the Transmission Control Protocol (TCP) networks using a public key called X.509 certificate. If an X.509 certificate is very weak it can be cracked by the attackers. For this reason, the system cryptography used to build certificates must be robust to prevent most attacks (such as Man in the Middle MITM). To decrease the weakness, the Operation Systems need robust key random generators. The longer the key, the more difficult is to break in. For example, a 1024 bits Rivest Shamir and Adelman (RSA) is more vulnerable than a 2048 bits RSA. If attackers obtain the private key, they can listen TLS traffic and decrypt it. There are several methods useful to prevent attacks. For example, there are different alternatives to the RSA keys, such as Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) [4]. Meanwhile, this can be mitigated using Perfect Forward Secrecy (PFS), a property ensuring a security transmission so that if a long-term key is compromised, the session key derived from it is safeguarded. Therefore Heartbleed is an important implementation vulnerability different from other attacks, such as Crime, Beast and Breach [12]. In this case, a programming issue occurred in Open SSL which generated implementation vulnerability in Transport Layer Security (TLS) protocol heartbeat extension [7].

B. Cloud Computing

Cloud computing means to hire the services of cloud vender on pay as per use basis over the internet [15]. The

 ISBN: 978-1-61208-427-5

Cloud customer uses Web Browser to access the services that it rendered from Cloud vender. Although immense research has been carried out to find out the security challenges and issues on the Cloud but Cloud is not adequately secured as traditional IT computing. Secure connection between Cloud customer and Cloud vender is highly important which must be secured by the implementation of encryption and TLS/SSL [13]. The Cloud customer store and processes its highly sensitive data on the machines that it rendered over the Cloud network. All the data travels on internet from customer computer to Cloud Service Provider (CSP). There are lot of free cloud services available over the internet like a Dropbox, Google Docs, Flicker and etc. Most of the Cloud organizations use Open Secure Sockets Layer (OpenSSL) to provide a secure platform for transformation of data over the internet. Among Cloud security risks, the Heartbleed security vulnerability made attackers to breakdown Cloud encrypted communication by exploiting serious security vulnerability in OpenSSL library and get about 24-55% protected memory contents of the Cloud machines and ultimately attacker become successful to get significant amount of information. The Cloud organizations which have customers running TLS and not relying upon OpenSSL are not infected with Heartbleed vulnerability, but its ratio is very low. The research found a number of Cloud risks that are very harmful for the Cloud Websites. The Cloud risks are given in the Section V of this research paper. The authors keeping in view the Cloud risks, checked the Cloud websites of the Italy. The list of the websites that are infected due Heartbleed bug is not appended in the paper due to security concerns of these websites.

C. New vulnerabilities

Heartbleed is a 'server side' attack, therefore it can be conducted by a remotely attacker simply knowing the public IP address of the target. In this work we show also new web vulnerabilities as POODLE and FREAK. These vulnerabilities are 'Man in the Middle' (MIMT), therefore, their impact is only for clients and local network environment ('client side'). POODLE is a vulnerability of clients (e.g. web browsers) for the support of SSL 3.0 [11]. It allows an attacker to decipher 'SECURE' HTTP cookies on the local network. FREAK allows an attacker to intercept HTTPS connections (MIMT) between vulnerable clients and servers and force them to use weakened encryption[16]. At the website 'freakattack.com' it is possible to control the vulnerability of web-browsers and of a great dataset of vulnerable websites (servers) to FREAK.

The paper is organized as follows: in Section 2, we describe the heartbeat vulnerability and the relative Heartbleed attack. Section 3 presents a simulation of Heartbleed attack through Python Script Language. Section 4 present an own HTTPS Italian website dataset developed for this paper and the relative results obtained using Qualys SSL Labs online tool. Section 5 shows a detailed analysis about Cloud Risks of Heartbeat vulnerability. Finally, in Section 6 we discuss open challenges about HTTPS website security.

II. HEARTBEAT AND HEARTBLEED

Heartbleed attack is a bug present in OpenSSL versions 1.0.1 through 1.0.1f. It allowed attackers to steal and to analyze private cryptographic keys. The first reason that allowed this kind of attack is that there were not security checks in code that implemented TLS protocols. Attackers had access to memory space used by TLS to store data like session key, in the server. Therefore, attackers could handle traffic exchange from clients to server and vice versa, stealing password and other user's information. As indicated in [2], The affected versions of hardware/software were those which used vulnerable versions of OpenSSL. Some operating system distributions that have been shipped with potentially vulnerable OpenSSL version were Debian Wheezy (stable), OpenSSL 1.0.1e-2+deb7u4; Ubuntu 12.04.4 LTS, OpenSSL 1.0.1-4ubuntu5.11; CentOS 6.5, OpenSSL 1.0.1e-15; Fedora 18, OpenSSL 1.0.1e-4; Open BSD 5.3 (OpenSSL 1.0.1c 10 May 2012) and 5.4 (OpenSSL 1.0.1c 10 May 2012); FreeBSD 10.0 - OpenSSL 1.0.1e 11 Feb 2013; NetBSD 5.0.2 (OpenSSL 1.0.1e); Open SUSE 12.2 (OpenSSL 1.0.1c). It uses TLS protocol Heartbeat extension kept the channel communication alive when there were not information to exchange between end-users. For this reason this bug allowed stealing under normal condition all the information handled by the TLS encryption. The Heartbeat Message was exchanged using an SSL3 RECORD structure.
The Heartbleed mechanism consists of some phases[5]. The detail of these phases (Figure 1) is given below:
1. Potentially attackers, send any heartbeat messages request to device running a vulnerable version of OpenSSL. These simple message consists of two key fields: a payload length (64 KB) and data.
The structure is the following:

```
struct ssl3Record
{
int length;
char *data;
}
```

2. When the peer gets the message request, this is processed and the request is written to memory allocating a buffer for response;
3. OpenSSL copies the payload content into buffer allocated, without bounds checks, considered it trust;
4. OpenSSL returns a message response containing the original payload and other private information like long terms server private keys, session ticket keys, confidential data and TLS session keys.

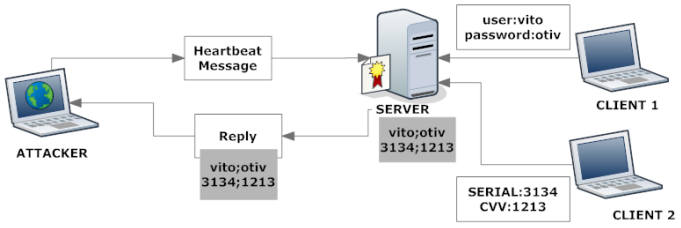


Figure 1. Heartbleed attack

Through this implementation vulnerability, the attackers were allowed to access data contained in the security infrastructure that used OpenSSL gathering personal information. Meanwhile it is not possible to know if someone had exploited this bug against our system architecture because this implementation vulnerability does not leave any trace in the logs. Today, there are different solutions adopted to face and to prevent this bug; the first is to update the OpenSSL version or alternatively, the OpenSSL code has to be recompiled remov-

 ISBN: 978-1-61208-427-5

ing from it the time option DOPENSSL_NO_HEARTBEATS.

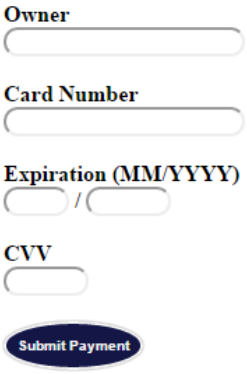


Figure 2. Payment Test Form

III. HEARTBLEED ATTACK SIMULATION

To understand the concrete impact of the Heartbleed, in this section, we show Heartbleed attack in a local environment. To conduct this attack, on Heartbleed vulnerability, we have considered a solution PYTHON based on the exploit code of eelsivart [15]. This solution considers a hostname as parameter and attacks the Heartbleed vulnerability. As case study we have considered a Windows Environment with XAMPP 1.8.3, APACHE 2.4.7, PHP 5.5.9 and OpenSSL/1.0.1e (known version vulnerable to Heartbleed). We have considered two PHP forms: a classic contact form and a classic payment one given in Figure 2 above. In this example, we can imagine a user that inserts information in two website forms protected by SSL. The output in Figure 3 shows the parameters of contact form extracted by Heartbleed attack. The parameters (name, email, subject, and message) of contatti.php are then visible to the attacker. The same output appears for the form credito.php that shows the credit card parameters inserted (Figure 4). Then, it is possible for an attacker to schedule distributed extractions to obtain a lot of information through Heartbleed. These extractions can be automated and elaborated by a simple text parser that catalogs only the relevant information. Consequently, this example underlines the dangerousness of Heartbleed attack that allows an attacker to retrieve a lot of useful information (e.g. usernames, passwords, credit cards).

```
C:\Python27>python.exe C:\testheartbleed.py 192.168.18.229

Connecting to: 192.168.18.229:443

HEARTBEAT WARNING: 192.168.18.229:443


User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/41.0.2272.101 Safari/537.36

Referer: https://127.0.0.1/form/contatti.php

nome=Domenico&email=domenico.dicarlo%40dicarlo.biz&oggetto=Informazioni+commerci
ali&messaggio=Salve%2C+vorrei+chiedere+informazioni+sul+prezzo+del+prodotto+n%C2
%B0535+presente+nel+vostro+listino.%0D%0ASaluti+Domenico&submit=Invia+messaggio|
....%.A.......d/+...6...6.8...8....P
```

Figure 3. Heartbleed test

```
C:\Python27>python.exe C:\testheartbleed.py 192.168.18.229

Connecting to: 192.168.18.229:443

HEARTBEAT WARNING: 192.168.18.229:443


User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/41.0.2272.101 Safari/537.36

Referer: https://127.0.0.1/form/credito.php

intestatario_carta=Vito+Santarcangelo&numero_carta=1717171717171717&mese=11&anno
=11&cvv=121&submit=Effettua+Pagamento..*mQ.;E..[...R,/+...6...6.8...8....P
```

Figure 4. Heartbleed test

IV. 2015 WEBSITE ANALYSIS

In this section we show the experiment conducted on an own dataset of Italian HTTPS website (.it extension). The dataset of websites has been built by us considering the result of the query ‘inurl:https:// .it’ on ‘www.google.it’ using the search operator ‘inurl:’, and categorizing the Italian websites in three groups: Bank, Institution, Other (e.g. Mail, Cloud, E-commerce, Bet Services, Billing). We have named this dataset as ‘IT-HTTPS-DB’. We can share it (by e-mail request at vito.santarcangelo@centrostudi.biz) for research purposes. Our database is made of 100 websites: 35 bank, 35 public institution and 30 other HTTPS websites.

We have conducted the vulnerability test with the use of SSL Labs online tool considering as metric their SSL Server Rating [9]. SSL Server Rating is a metric of Qualys SSL Labs that assigns a letter from A (strong) to F (weak) to evaluate the security level of a website.

The results obtained are that only the 1% is characterized by Heartbleed vulnerability, so the impact of this attack can be considered almost completely solved in only one year. Nevertheless, we can see that a lot of websites are affected by some important Man in the Middle (MIMT) vulnerabilities as POODLE (43%) and FREAK (20%). Heartbleed or POODLE over TLS or FREAK defines F valuation, POODLE over SSL3 defines C valuation[14]. Considering the SSL Server Rating the results show that the 35% of websites examined has F valuation, the 20% has C valuation, the 31% has B valuation and only 14% has A valuation.

The alarming results are that no website of public institutions has been valued as A and the 54% of them has a value of F. Also even more serious is the fact that the 25% of Italian bank websites has a valuation between C and F. The distribution of the valuations given in Table I for the three clusters considered (banks, institutions, others).The Table II shows the impact of Heartbleed, Poodle and Freak vulnerabilities on the three website clusters [11]. An important feature extracted is that 50% of the websites considered uses Apache engine and the 21% Microsoft IIS one.

TABLE I. SECURITY ASSESSMENT

	Metric			
	[A]	[B]	[C]	[F]
Banks	10	16	5	4
Institutions	0	6	10	19
Others	4	9	5	12
Total	14	31	20	35

During the test it has been possible to identify also other problems for these websites (2% of banks and 31% of public institutions). In fact some of them are characterized by the use of the SSL2, MIMT Insecure Renegotiation, OpenSSL CVE20140224 and Diffie-Hellman (DH) key exchange parameters.

 ISBN: 978-1-61208-427-5

TABLE II. VULNERABILITY ASSESSMENT

	Vulnerability		
	[HEARTBLEED]	[POODLE]	[FREAK]
Banks	0	6	2
Institutions	0	25	13
Others	1	13	5
Total	1	43	20

Another interesting test has been conducted by our team on PDA (point of access) for the Italian telematics process [8]. We have considered a dataset of 55 PDA clustered as Public Institutions, bar associations and Privates. The results shows that, despite the importance of the information exchanged through these channels, no website has obtained the A valuation, only 1 website (private) has obtained B valuation, 42 websites have obtained C valuation and 8 websites have obtained F valuation. For 4 websites it has not been possible to test the metric as not accessible. No website is affected by Heartbeat vulnerability, however, 89% of PDA is affected by POODLE vulnerability and 7% by FREAK. 100% of public PDA examined has a valuation between C and F (see the Table III below).

TABLE III. PDA SECURITY ASSESSMENT

	Metric			
	[A]	[B]	[C]	[F]
Bar Assoc.	0	0	39	0
Public	0	0	2	3
Private	0	1	1	5
Total	0	1	42	8

V. HEARTBLEED AND CLOUD RISKS

The vulnerability of the Heartbeat in OpenSSL can cause the following cloud risks and the severity level of these risks is very high.

A. Network Failure

The attacker can fail the internal CSPs switching and routing network may endanger connectivity for Cloud customer environments. Furthermore, the CSP may lose its control on external network connections. External connectivity is a critical part of the services offered. Data can be lost / damaged or network storage may be prohibited. Similar type of problem may occur if meta-information about data is lost.

B. Distributed Denial of Service (DDOS)

The Heartbleed vulnerability shall provide Cloud Server information which can be used to Distributed Denial of Service (DDOS) attacks on IP addresses within the network can easily harm the services of Cloud.

C. Loss of Customer Account and Configuration Data

Account settings and configuration data are essential in the process of service delivery. As mentioned above about the vulnerability, it can cause memory contents loss of customer accounts and configuration data can result in loss of service.

D. Data Interception

By exploiting Heartbleed vulnerability the attacker can intercept the data of the Cloud server machine. This situation will make Cloud Computing more vulnerable to attacks such as replay attacks, man-in-the-middle, spoofing, eavesdropping and sniffing.

E. Theft of Data

The attacker can theft the data of the Cloud users and he/she does not know what is going on behind the scene and they just suppose that they are transferring data to a secured Cloud Service Provider and their data is not intercepted.

F. Loss of Encryption Keys

The attacker could get private keys that sites use to encrypt and decrypt sensitive data. These keys are further used to encrypt all the traffic between Cloud customer and Cloud Service Provider. The attacker can get passwords of usernames and actual contents of data. This risk can further cause eavesdropping and theft of data.

G. Unauthorized Access

The active intruder after attacking on Cloud Service Provider website can gain unauthorized access to CSPs server machine and it may exploit the integrity and privacy of the customers. Fake users gain access to restricted areas.

H. Business Continuity

Due to Heartbleed attack, Cloud services can be blocked and Cloud customer may not become able to access it data over the Cloud network. The Cloud customer will also bear its financial loss as well as its business continuity will also be affected.

It is further added that during literature review, it is studied that Cloud Security Alliance in its report published on April 10, 2014 stated that after 24 hours of Heartbleed vulnerability discovery, 368 Cloud organizations were still vulnerable[10]. The Skyhigh recommended to the Cloud organizations to update OpenSSL and obtain new certificates. Furthermore, CSA has recommended five steps that every Cloud organization have to take in case infected Heartbleed.

VI. CONCLUSION AND FUTURE WORK

This paper has shown methods and tools to test the Heartbleed vulnerability. The results obtained by python code shows the importance and dangerousness of this vulnerability. Luckily, considering our dataset of 100 HTTPS public and private sector websites including Cloud websites and find out that only 1% of the websites is still affected by this vulnerability, meanwhile, POODLE and FREAK vulnerabilities are the new security problems to vanquish. An alarming scenario is that of the four banks characterized by POODLE, FREAK and MIMT vulnerabilities and of PDA (point of access) for Italian telematics process that shows as a better sensitivity about IT security problems is required. Interesting open topics to implement in future works are the extension of the analysis to other countries to compare the Italian results, the re-monitoring of these website farther on, a detailed analysis on POODLE and FREAK and a review on Shellshock attack.

 ISBN: 978-1-61208-427-5

REFERENCES

[1] Z. Kasten, D. Adrian, J. Halderman, and M. Bailey, 'The Matter of Heart-bleed', In Proceedings of the 2014 Conference on Internet Measurement Conference, ACM, pp. 475-488, 2014.

[2] M. Mshangi, 'Using Soft Systems Methodology and Activity Theory to Exploit Security of Web Applications against Heartbleed Vulnerability', International Journal of Computing & ICT Research, 8(2), pp. 32-52, 2015.

[3] J. A. Lewis, 'Heartbleed and the State of Cyber Security', Taylor and Frances Group, pp. 294-299, 2014

[4] E. Dreyfus, 'TLS hardening', BSD Magazine, 2014, [Retrieved on May, 2015].

[5] B. Chandra, 'A technical view of the OpenSSL Heartbleed vulnerability', IBM White Paper,2014, [Retrieved on May, 2015].

[6] US Department of Homeland Security, 'Heartbleed OpenSSL Vulnerability', National Cyber Security and Communications Integration Center, 2014, [Retrieved on May, 2015].

[7] Websense, 'OpenSSL Vulnerability CVS-2014-0160 (Heartbleed)', Web-sense White Paper, 2014, [Retrieved on May, 2015].

[8] V.Santarcangelo, G.Oddo, N.Santarcangelo, V.Ribaudo, and G.Lamacchia, 'Fattura elettronica, conservazione sostitutiva, processo telematico: Stru-menti per il miglioramento della qualita nella pubblica amminis-trazione', Sei Sigma e Qualita, RCE Multimedia vol.5, n.4 (2014).

[9] Qualys SSL Labs, 'SSL Server Rating Guide', 2014, [Retrieved on May, 2015].

[10] H. Byun, '24 Hours After Heartbleed, 368 Cloud Providers Still Vulnerable', CSA Security Alliance, 2014.

[11] B. Mller, T. Duong, and K. Kotowicz, 'This POODLE Bites: Exploiting the SSL 3.0 Fallback', Google Security Advisory, 2014.

[12] T. Duong and J. Rizzo, 'Here Come The Ninjas', 2011, [Retrieved on May, 2015].

[13] M.I. Tariq, 'Towards Information Security Metrics Framework for Cloud Computing', IJ-CLOSER, (2012).

[14] ISE, 'FREAK Security Advisory' , ISE CONFIDENTIAL, 2015, [Retrieved on May, 2015].

[15] Eelsivart, https://gist.github.com/eelsivart/10174134 , (2014), [Retrieved on May, 2015].

[16] M. R. Albrecht, D. Papini, K. G. Paterson, and R.Villanueva, 'Polanco Factoring 512-bit RSA Moduli for Fun (and a Profit of $9,000)', 2015, [Retrieved on May, 2015].

 ISBN: 978-1-61208-427-5

A Detection and Prevention Algorithm for Single and Cooperative Black hole Attacks in AODV MANETs

Saeed K. Saeed **Noureldien A. Noureldien**

Department of Computer Science

University of Science and Technology

Omdurman, Sudan

e-mail: saeed_kl@hotmail.com **e-mail: noureldien@hotmail.com**

***Abstract*— A mobile ad-hoc network (MANET) is a new generation of wireless networks that is used in many applications. MANETs have much vulnerability such as mobility, unsecure boundaries, lack of central management, that have been exploited by attackers to launch different types of attacks. One well known attack is the Black Hole Attack, which absorbs packets before reaching its destination. As one of the vital MANET attacks, the black hole attack has been studied extensively, and many detection and prevention techniques have been proposed. In this paper, a new detection and prevention algorithm for single and cooperative black hole attacks in MANET that employ Adhoc On-demand Distance Vector (AODV) is proposed. The developed algorithm benefits from the two previously proposed detection techniques; the sequence number scheme, and cooperative black hole attack scheme in AODV MANETs. The simulation results show that the proposed algorithm works and improves the security of AODV MANTETs against black hole attack.**

Keywords- MANET attacks; Black hole attack; Black hole attack detection; single Black hole attack; Cooperative black hole attack.

I. INTRODUCTION

MANET's are composed of equivalent nodes that communicate over wireless links without any central control and can move randomly and have the capability to self-manage without any need to predefined infrastructure.

The nodes can cooperate to communicate with each other via sending data packets from source to destination through intermediate node(s). Packet routing is done using a routing protocol such as AODV, which is the most popular routing protocol.

MANET's are facing great security challenges due to the vulnerabilities initiated from; the wireless transmission media, the high dynamic topology of nodes, the limited nodes resources, and the lack of central management.

These attacks can prevent the transmission or reduce the performance of the network. One of these attacks is the Black Hole Attack in which one or more malicious node drops all the data packets in the network. As a result the data packets do not reach the destination node and the data will be lost.

To defend a black hole attack there are a lot of techniques that have been proposed either to detect or to prevent black hole attack.

In this paper, we discuss some of the most common current detection techniques with a special focus on the sequence number scheme and the detection of cooperative black hole attack scheme, and then a new algorithm that is capable for detection and prevention of single and cooperative black hole attack is proposed.

The rest of this paper is organized as follows: Section 2 reviews the current black hole attack detection techniques. Section 3 is dedicated to the proposed new algorithm. Section 4 shows the simulation results of the proposed algorithm. Section 5 represents conclusion and future work.

II. CURRENT DETECTION TECHNIQUES

There are many secure routing protocols, and schemes, which can detect the black hole attack; this section discusses some of these techniques.

A. Neighborhood-based and Routing Recovery Scheme

Guan et al. [1] designed a method to deal with the black hole attack based on the neighbor set information; this method consists of two parts: detection and response. The detection procedure has two major steps; in the first one each node collects neighbor set information. The second step determines whether if there is a black hole attack or not. In response procedure, the source node sends a control packet called Modify Route Entry (MRE) to the Destination node in

 ISBN: 978-1-61208-427-5

order to form a correct path by modifying the routing entries of all of the intermediate nodes from source to destination.

B. Detection Based on Path Based Method

A path based scheme is proposed in [2]. In this method, a node is used to monitor the next hop nodes in the current route path. First, the monitoring node calculates the digest value for every packet that wants to be sent, and add this digest into a buffer called FwdPktBuffer. After sending the packet the node overhears, when the next hop forwards this packet that is overheard, the digest value will be released from the FwdPktBuffer. Finally, every node calculates the forwarding rate of its next hop and compares it with a threshold. If it is lower than the threshold, that node is marked as malicious.

This technique does not increase the overhead because it does not send additional control packets and also it does not require encryption of the control packets to avoid the security primitive attacks.

C. Detection Based on Learning Automata

Taqi and Abdorasoul [3] proposed a black hole attack detection mechanism that uses a machine learning automata is proposed. The machine operates in a random environment and tries to adapt itself to this environment according to feedback received from this environment. The machine has a finite set of potential actions, where each action has a specific probability. This probability is updated according to feedbacks. The feedbacks may reward or punishment. If the machine performs an action in the correct manner it will get rewarded, otherwise it will get punished. Action probabilities affect the selecting of the future action. The main objective of this is that automata should learn how to select the best action from the finite set of actions. Therefore, the best action is the one that maximizes the probability of getting reward from the environment.

Each node has a list of its direct neighbors and gives a value of trust and a confidence degree to each one of those neighbors. The initial value of trust is 1. This means each node in the network is trust in all of its neighbors. So any node has normal behavior in the network. The value of the trust will be updated after receiving feedback from the network. Each node dedicates a learning automaton to compute the degree of confidence of each neighbor. According to this degree, the node will decide if it will send packets through that node or not.

D. Detection Using Fuzzy Logic

Jagpreet [4] have proposed a system that isolates the malicious node from the network. Every node in the network decides if the behavior of its neighbors was malicious or not. If a node decides a neighbor is malicious it will broadcast an alarm packet in the network with the IP address of the malicious node which is not allowed to participate in any future communication.

The fuzzy system integrates with AODV routing protocol. It consists of four components: Fuzzy Parameter Extraction, Fuzzy Computation, Fuzzy Verification Module and Alarm Packet Generation Module. The Fuzzy Parameter Extraction module extracts the required analysis parameters from the network traffic. Then pass these parameters to a fuzzy computation module, which in turn applies some of fuzzy rules and membership functions to calculate the fidelity level of the node. The verification module determines the behavior of the node by comparing the value of fidelity level with the threshold if it was less than the threshold level in fuzzy it will broadcast an alarm packet with the IP address of this malicious node to the whole of the network. This system beside the detection of the black hole, it also isolates it from the network.

E. Detection Using Anomaly Detection

Fantahun, and. Zhao [5] proposed a host-based Intrusion Detection System (IDS) scheme. The scheme assumes that, every activity of a system can be monitored and anomaly activities of an intruder can be identified from normal activities. Hence, by identifying anomalous activities of an adversary, it is possible to detect a possible intrusion and isolate the adversary. To do so an anomaly detection system needs to be provided with a pre-collected set of anomalous activities, called audit data.

Once the audit data is collected and given to the system, the system will be able to compare every activity of a host with the audit data on the fly. If any activity of a host resembles the activities listed in the audit data, the anomaly detection system isolates the particular node by forbidding further interaction. It does not trust on peer nodes.

F. Enhance Black-Hole AODV (EBAODV)

Rachh et al. [6] proposed an Enhance Blackhole AODV solution (EBAODV). In this solution, what is called leader nodes are created first, these nodes are responsible for detection of malicious nodes.

After sending the first Routing Request message (RREQ) a timer is started. If a RREP is received before the timer is expired, then one stale packet will be send to the destination. To ensure that the stale packet is received by the destination, the source node must received acknowledgement (ACK) from the destination. When the source node receives the acknowledgement it sends the original packet.

When the source node has not received any ACK, it means packets are dropped. If the number of dropped packets is more than a threshold, then the leader nodes will send block messages that contains the id of the blackhole node to all neighbors and the source node must start again a new RREQ to discover another route.

G. Feedback Solution

Singh [7] proposed a feedback technique which examines the malicious nodes from the amount of packets sent by this

 ISBN: 978-1-61208-427-5

node, this amount in the most of the cases equal zero. After detecting the malicious nodes, the method was adopted to avoid the recipients of the packets that are coming from these detected nodes. The packets coming to the neighbors of the black hole nodes are propagated back to the source, and the source node has to follow another route to the destination. This method decrease packet loss in the network comparatively. But this method cannot detect the collaborative black holes.

H. Detection Technique for Single Black hole Attack Using Sequence Number

Singh and Manpreet [8] have proposed a method to find the secure route and prevent the black hole nodes (malicious nodes) in MANETs. This is done by checking whether there is a large difference between the sequence number of the source node and the intermediate node who has sent back the first RREP or not.

The detection method builds a table to store RREPs messages received in response to source RREQ. The method compares the sequence number in RREPs with that of the resource node, if there is a significant difference, the method considers that RREP is originated from a malicious node and remove it from the table. The RREP with a reasonable difference is considered to be from a legitimate node and the route defined by that RREP is used by the source node.

The method was implemented by adding a new function to AODV protocol called Pre_ReceiveReply (Packet P) and added a new table C_RREP_T, a timer M_WAIT_T and a variable M_Node to the data structures in the basic AODV. The time M_WAIT_T is initialized to be the half value of RREP_WAIT_TIME, i.e., the time for which the source node waits for RREP control messages before regenerating RREQ.

The source node analyses all the stored RREPs from C_RREP_T table and discards the RREPs having very high destination sequence number. Then the source node selects a reply having highest destination sequence number of the C_RREP_T table.

The major drawback of this method is that; when the source node received RREPs from two or more collaborated malicious nodes, then the function will fail to get a legitimate RREP since the collaborative attackers keep sending similar sequence numbers. So this technique fails to detect cooperative black hole nodes.

I. Detection Technique of Cooperative Black Hole Attack

Munjal et al. [9] proposed a method to detect multiple black hole nodes that working collaboratively as a group to launch a cooperative black hole attack. The technique maintains a Routing Information Table (RIT) at each node in addition to the Routing Table of the AODV protocol. The method considers a node that has an entry in the RIT table as a trusted node. RIT table contains the fields {Node ID, From, Through}, where From Node stands for source nodes that broadcast RREQs, and Through Node stands for nodes that forward data packets.

The technique suggests that each source node builds a table for trusted nodes that are exchanged RREQ or data packets with the source node.

The source node starts to send RREQ and then wait for destination replies, all intermediate nodes update their RITs by adding an entry referred to the destination. Also the source node updates the RIT as well. When a node receives a RREP message, it checks its Trust table, if the sending node was recorded as a trustee the RREP is accepted, otherwise the source node makes a further request (FRREQ) to its neighbors.

This technique leads to a delay as a result of trust table checking and exchanging of further requests and further replies.

J. BDSR Scheme to Avoid Black Hole Attack

Po-Chun et al. [10] designed a novel solution named Bait DSR (BDSR) or Fake RREQ scheme to prevent the collaborative black hole attacks. The proposed mechanism is composed of proactive and reactive method to form a hybrid routing protocol, and the major essence is the DSR on-demand routing. In this solution in the beginning of routing phase, the source node sends bait RREQ packet before starting route discovery.

The target address of bait RREQ is random and non-existent destination. The malicious nodes are easily expelled from the initial phase, because the bait RREQ is able to attract the forged RREP from the black hole node. In author's mechanism, the generator of RREP is recorded in the RREP's additional field. Therefore the source node can recognize the location of an attacker from the reply location of the RREP. All of the response set by the adversaries should be dropped. After the initial phase, the authors employ the original DSR route discovery procedure. If the data delivery rate is lower than the pre-defined threshold value, the boot procedure will be triggered again to examine the uncertainly suspicious nodes.

K. Detection Using Watchdog

Marti et al. [11] proposed a method that uses the node promiscuous mode. This method allows a node to intercept and read each network packet that arrives in its entirety. Promiscuous mode means that if a node A is within the range of node B, it can overhear communication to and from B even if those communications do not directly involve A.

The watchdog works as follows, node A listens to the transmission of B to make sure that B forwards to C. If the packet overheard from B matches that stored in the buffer, it means that B really forwards to the next hop, and it then removes the packet from the buffer. However, if there's no matched packet after a certain time, the watchdog increments the failures counter for node B. If this counter exceeds the threshold, A concludes that B is misbehaving and reports to the source node S. The watchdog is implemented by

 ISBN: 978-1-61208-427-5

maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for the forwarding of the packet. If the tally exceeds a certain threshold, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. The watchdog technique has advantages and weaknesses.

A Watchdog has the advantage that it can detect misbehavior at the forwarding level and not just the link level. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions, limited transmission power and false misbehavior alarm.

L. Detection Based on Collaborative Bayesian Watchdogs

Authors in [12], have proposed a detection technique based on the message passing mechanism between a group of collaborating Bayesian watchdogs by allowing to every watchdog to publishing both self and neighbor reputations. The standard watchdog monitors packets have been transmitted/ received by its neighbors, counts the packets that need for retransmission, and calculates the trust level for any one of the neighbors as the ratio of packets retransmitted to packets that need to retransmission. If a node retransmits all the packets that it should have retransmitted, it will be given the value 1 as a trust level. If there is a node has a trust level lower than the tolerance threshold, the watchdog will consider this node as a black hole.

III. A PROPOSED NEW DETECTION AND PREVENTION ALGORITHM FOR BLACK HOLE ATTACKS IN AODV MANETS

Our proposed black hole detection and prevention technique is based on the sequence number scheme [8] and cooperative black hole attack scheme in AODV MANETs [9].

The Sequence Number Scheme as explained in section II.H, suffers from collaborative black hole attack; whiles the proposed technique for detecting cooperative black hole attacks, discussed in section II.*I* causes a high overhead over the original AODV.

Our proposed detection technique aimed to take advantages of the two schemes and to avoid their drawbacks, to be able to detect single and cooperative attackers without significant delay over normal AODV.

The algorithm donates a table for each node to store received RREPs after the node sends a RREQ, besides a table for trusted nodes. Then after, the source node scans the RREPs table looking for a trusted node that is registered in the trusted nodes table. If such a node is found, then the route defined by that RREP is used.

If the scan fails to find a trusted node, then the sequence number scheme applies to RREPs table entries. Entries with very high sequence numbers will be deleted. If a RREP with an adequate sequence number is found, then the RREP source node is considered a trusted node and added to trusted table and the route defined is used.

Thus, the proposed algorithm defines the following pre-setting:

- At each node a W-TIME timer was set.
- At each node an RREP_TABLE was built to store received RREPs during W-Time.
- At each node a TRUST-TABLE was built to store trusted nodes.

Then, the algorithm works as follows:

1) The source node sends it's RREQ and wait for W-TIME, storing all received RREPs in RREP_TABLE.

2) IF a RREP from a node in the TRUST_TABLE is found in RREP_TABLE, use that RREP route, then call normal AODV and terminates;

3) Otherwise While RREP_TABLE is not empty do

a) If a RREP have a very high sequence number, then delete the RREP route from RREP_TABLE // applying sequence number scheme

b) IF a RREP have a suitable sequence number, then add that RREP node for the TRUST_TABLE, use that RREP route, then call normal AODV and terminates;

4) IF RREP_TABLE is empty go to step 1

This algorithm applies both the Sequence Number Scheme in step 3.a, and the Trust Table technique used in the detecting cooperative black hole attack method in preprocessing stage and in steps 2 and 3.b.
Step 3.a and Step 5, ensures the avoidance of cooperative black hole attack.

The developed algorithm was implemented by adding the TRUST_TABLE, RREP_TABLE, a timer W_TIME and a Boolean variable NOT_ROUTE to the data structures in the basic AODV.

To implement the algorithm a new function to AODV protocol called Pre_ReceiveReply (Packet P) is added in aodv.cc before ReceiveReply (Packet P).

The pseudo code of the new Pre_ReceiveReply (Packet P) function is shown in Figure 1.

 ISBN: 978-1-61208-427-5

```
Pre_ReceiveReply (Packet P)
  {
    While (NOT_ROUTE = true) do
     {
        Send RREQ;
        While (W_TIME)
                Store all received RREPs in
                 RREP_TABLE;
       i=0;
       While ( RREP_TABLE is not empty)
         If (RREP_TABLE[i] is in TRUST_TABLE
         {
            NOT_ROUTE = False;
             Use that RREP route;
             Call normal AODV;
             Exit;
           }
          Else
           i++;
       }/**while
      i=0;
       While (RREP_TABLE is not empty)
       {
           If ( RREP_TABLE[i].Dest_Seq_no >>>
           Src_Seq_No) then
                 {
                   delete(RREP_TABLE[i]);
                   i++;
                 }
           else
            {
               Add RREP_TABLE[i]node to the
                        TRUST_TABLE,
              NOT_ROUTE = False
               use that RREP route,
               call normal AODV;
               exit;
             }
        }//while
    }
}
```

Figure 1. Proposed Algorithm

The Pre_ReceiveReply (Packet P) function implements our proposed detection algorithm.

IV. SIMULATION AND RESULTS

To test the performance of the developed algorithm, three scenarios are simulated. The first scenario simulates the network under the normal AODV (called Normal-AODV), the second scenario simulates the network under both single and cooperative black hole attacks (called Blackhole-AODV), and the last scenario simulates the network that implements proposed algorithm (called Modified-AODV-black hole).

The Network Simulator (NS2.35) was used as a network simulation tool. The Tool Command Language (TCL) was used to implement the simulation with 25 mobile nodes. All simulation time is set to 100 Sec. Table (I) shows the simulation environment that used in all experiments scenarios.

TABLE 1: THE SIMULATION ENVIRONMENT

Simulator	NS-2 v. 2.35
Transmission Protocol	UDP
CBR Packet Payload (data)	1000 bytes
Channel bit rate (data)	20 Mbps
Number of nodes	25 nods
Routing Protocol	AODV
Traffic Model	CBR
Terrain	1186 x 584 meter
Malicious nodes	3 nodes
MAC type	802.11
Simulation Time	100 Sec

To evaluate the performance of the developed algorithm, packet delivery ratio and throughput are used as measurement criteria. To represent and illustrate results the Xgraph tool is used.

Figures 2 and 3 show the simulation results. Where the green color represents the Normal-AODV, the red color represents the Blackhole-AODV, and the blue color represents the Modified-AODV-black hole.

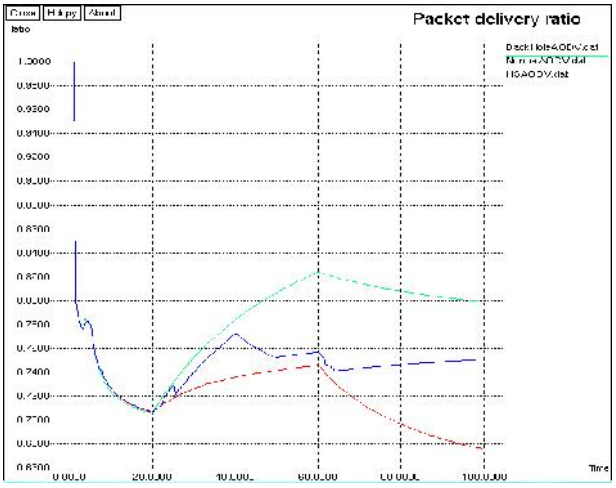


Figure 2. Throughput

 ISBN: 978-1-61208-427-5

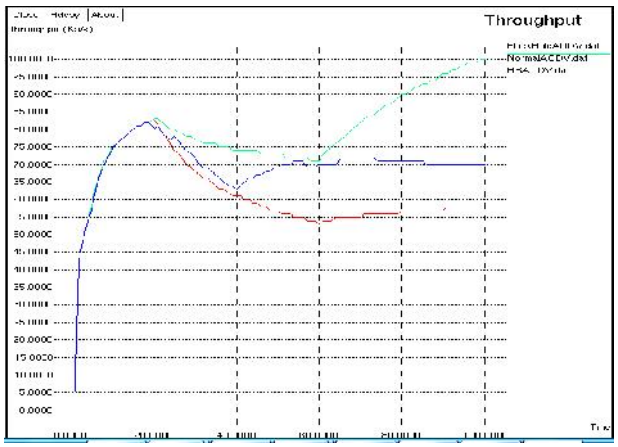


Figure 3. Packet Delivery Ratio

A. Discussion

From Figure 2, the throughput of Modified-AODV-black hole started normally, then the performance decline due to the start of the developed algorithm the waiting period in which it receives and stores RREPs. The algorithm then executes making the performance moderate between the Normal-AODV and the Black Hole-AODV.

Figure 3 represents the packet delivery ratio (PDR) versus Time. The performance is similar to throughput. The PDR in the proposed algorithm started normally as well, then decreased because the source node might have just one trusted node or no trusted nodes in the TRUST_TABLE. Then, the Modified-AODV-blackhole performance moderates between the Normal-AODV and the Blackhole-AODV. This means that, the proposed algorithm enhance and improve the performance of the network under black hole attack.

V. Conclusion and Future Work

In this paper, a new detection and prevention algorithm to single and collaborative black hole attack was developed and tested. The simulation results show that the developed solution improves the security and resistance of MANETs to single and collaborative black hole attack.

Currently, we are working on comparing the performance of the developed algorithm with the previously proposed techniques, the sequence number and cooperative black hole attack schemes.

References

[1] S. Guan, J. Chen, and U. Pooch, “Detecting Black-hole Attack in Mobile Ad Hoc Networks,” Proc. the 5th European Conference on Personal Mobile Communicationse,, Apr. 2003, pp. 490-495, doi: 10.1049/cp:20030303.

[2] J. CAI, Y. Ping, C. Jialin, W. Zhiyang, and L. Ning, “An Adaptive Approach to Detecting Black and Gray Hole Attacks in Ad Hoc Network,” Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, 2010, pp. 775-780, doiI:10.1109/AINA.2010. 143.

[3] M. Taqi, and G. Abdorasoul, “Detecting Black Hole Attack in Wireless Ad Hoc Networks Based On Learning Automata,” Proc. the 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), Nov. 2011 , pp 514 – 519.

[4] K. Jagpreet, “Fuzzy Logic based Intrusion Detection System against Blackhole Attack on AODV MANET,” International Journal of Computer Application (IJCA), Special Issue on Network Security and Cryptography (NSC), 2011. pp. 28-35, doi::10.5120/4331-024.

[5] Y. Fantahun, and X. Zhao, “Preventing Black hole Attack in Mobile Ad-hoc Networks using Anomaly Detection,” Proc. International Conference on Future Computer and Communication, Wuhan, May 2010, pp. 672-676, doi: 10.1109/ICFCC.2010.5497455.

[6] A. Rachh, V. Yatin, and R. Tejas, "A Novel Approach for Detection of Blackhole Attacks," IOSR Journal of Computer Engineering (IOSR-JCE), Mar-Apr. 2014 . vol 16, issue 2, pp. 69-74.

[7] H. Singh, "An approach for detection and removal of Black hole in MANETS," International Journal of Researh in IT& Management (IJRIM), June 2011. Vol 1, issue 2, pp. 78-87.

[8] H. Singh and S. Manpreet,"Securing MANETs Routing Protocol under Black Hole Attack," International Journal of Innovative Research in Computer and Communication Engineering, June 2013 . vol 1, issue 4, pp. 808- 813.

[9] K. Munjal, V. Shilpa, B. Aditya, "Cooperative Black Hole Node Detection by Modifying AODV," International Journal of Management, IT and Engineering (IJMIE), August 2012. Vol 2, issue 8, pp. 484-501.

[10] T. Po-Chun, J. Chang, Y. Lin, H. Chao, and J. Chen, “Developing a BDSR Scheme to Avoid Black Hole Attack Based on Proactive and Reactive Architecture in MANETs,” Proc. the 13th international conference on Advanced Communication Technology, Seoul, 2011, pp. 775-780.

[11] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” Proc. the 6th International Conference on Mobile Computing and Networking (MobiCom’00), 2000, pp. 255-265, doi:10.115/ 345910.345955.

[12] M. Serrat-Olmos, H. Enrique, C. Juan-Carlos, T. Carlos, and M. Pietro, “Accurate Detection of Black Holes in MANETs using Collaborative Bayesian Watchdogs,” Proc. Wireless Days (WD), IFIP International Conference, 2012, pp. 1-6, doi:10.1109/WD.2012.6402811.

 ISBN: 978-1-61208-427-5

A Brief Survey of Nonces and Nonce Usage

Geir M. Køien

Faculty of Engineering and Science
University of Agder
Grimstad, Norway
Email: geir.koien@uia.no

***Abstract*—Last year the European Union Agency for Network and Information Security (ENISA) published a report on cryptographic protocols. A main verdict was that we still have not reached maturity for the design and analysis of cryptographic protocols. This is bad news for a society that has become dependent on well-functioning information- and communication technology (ICT) infrastructures. In this paper, we address this by investigating the *nonce*. The nonce, a number-used-once, is but one small element of a cryptographic protocol. That is, a small, but nevertheless a critically important element. Yet, there is relatively little to be found in the literature regarding the properties of the nonce. This is thus an attempt to improve on this, by providing an initial analysis of nonces and classifying types of nonces used in different cryptographical protocols.**

Keywords–Number-used-once; Nonce; Randomness; Freshness; Timeliness; Uniqueness; Non-repeatability; Cryptographic protocol.

I. INTRODUCTION

A. Background

The ENISA report "Study on cryptographic protocols" [1] published in 2014 investigates the state-of-the-art for cryptographic protocols. The conclusion is clear and points out that cryptographic protocols are not well understood:

> Whilst the security of basic cryptographic building blocks, such as primitives and protocols, is well studied and understood, the same cannot be said of cryptographic protocols. The scientific study of such protocols can be said to be still not mature enough.
>
> *– from the executive summary of [1]*

Also, we have the ENISA report "Algorithms, key size and parameters report 2014" [2]. This is an important report, in a yearly series, as it represents the nearest thing to consensus about state-of-the-art in cryptographic algorithms and levels of protection. Here we learn, amongst others, that when using a nonce as an Initialization Vector (IV), there are several traps to fall in. Section 4, "Basic Cryptographic Schemes", goes into detail about concerns with using nonces as IVs for block cipher modes and in hash-based schemes. The worries are mainly about non-randomness and predictability. Yet, the report is surprisingly vague about what a nonce really is:

> Many modes make use of either a nonce or an random IV. A nonce is a number used once, it is a non-repeating value but not necessarily random. Thus a nonce could be a non-repeating sequence number. On the other hand a random IV should be random, and unpredictable to the adversary.
>
> – from Section 4.1 in [2]

Nonces are an important part of many, if not most, cryptographic protocols. They do serve different purposes and the requirements on the nonces are not always explicit, making it hard to determine exactly what properties the nonces must have and to verify that they indeed to have the required qualities.

B. Randomness, Computationally Infeasible, The Birthday Paradox and Collisions

We shall use the terms randomness and pseudo-randomness. Actual bit-field randomness in a nonce information element (IE) is not always necessary or even desirable. For our purpose, we want the term merely to mean that the value of the "random" IE is uniformly distributed over the whole value range and that a priori guessing the value of a random IE is computationally infeasible. That is, even when knowing very large series of prior values, the adversary shall not have gained any practical advantage in guessing the next value.

In this paper, we will use the term *computationally infeasible*. We intend it to mean that it will not, under any circumstance, be practical for an adversary to brute-force guess the random IE. That is, pre-computation attacks of dictionaries and rainbow tables must be utterly impractical and beyond the reach of even the most powerful adversary. This implies that the range of IE values must be so large that pre-computation indeed becomes impractical.

The birthday paradox is well known in cryptography (Section 2.1.5 in [3]). In our context, we shall worry about it's application to collisions. Nonce collisions will, in some cases, be a severe security problem. The collision-free property may therefore be required. It is well known that collision probability for a random draw is roughly $2^{N/2}$, where N is the bit-field size of the random IE. This means that one will need the random IE to have a bit-field size of 2 x N. It seems generally accepted that "128 bit complexity" is enough to provide the *computationally infeasible* property. If collisions are a security problem, then one need a 256 bit field for the IE in question in order to reach the required security level. On the other hand, if collisions are not a worry, then a 128 bit nonce should suffice.

II. BRIEF LITERATURE SURVEY

There is a considerable body of literature concerning security protocols and cryptographic methods available. In this paper, we have focused mostly on investigating nonce usage as described in security protocols. That is, our focus is from a "user's perspective". In this respect we have investigated some classic security papers, some papers concerning formal verification of security protocols and some papers that explicitly mention and discusses nonces.

 ISBN: 978-1-61208-427-5

A. Older Literature

Nonces are a very important part of BAN logic, as reported in in the seminal research reports "A Logic of Authentication" [4]. In [4], there is substantial emphasis on the *freshness* as property of a nonce.

Nonces are also explicitly mentioned in "Prudent Engineering Practice for Cryptographic Protocols" [5]. Here, the authors clearly relate authentication *challenges* with nonces. We here also find a description of what a nonce may be:

- Timestamps
- Serial numbers (which must be *recent* somehow)
- Random numbers

These three examples are also found elsewhere in the literature. In [5], the authors also a provide a principle for nonces:

> Be clear what properties you are assuming about nonces. What may do for ensuring temporal succession may not do for ensuring association – and perhaps association is best established by other means.

We concur with the above principle (`Principle 6`).

The paper "A Survey of Authentication Protocol Literature: Version 1.0" [6] is not about nonces per se, but nonce use is mentioned frequently throughout the survey.

B. Books Covering Nonces

In the book "Handbook of Applied Cryptography" (HAC) [3] we find a fair amount of material on nonces. Here the authors have explicitly highlighted the time-variant properties. They put emphasis on the role nonces play in preventing replay attacks. This is of course in line with the freshness property of [4]. We furthermore note that the authors highlight the following (Section 10.3.1 [3]):

> The term nonce is most often used to refer to a random number in a challenge-response protocol, but the required randomness properties vary. Three main classes of time-variant parameters are discussed in turn below: random numbers, sequence numbers, and timestamps. Often, to ensure protocol security, the integrity of such parameters must be guaranteed (e.g., by cryptographically binding them with other data in a challenge-response sequence). This is particularly true of protocols in which the only requirement of a timevariant parameter is uniqueness, e.g., as provided by a never-repeated sequential counter.

The "time-variant" property seems to be as much about ordering as about time per se. In this sense it is more concerned with uniqueness and freshness than about temporal properties.

In the book "Security Engineering" [7], by Anderson, we find only a few brief passages. This, otherwise quite comprehensive text, mentions that nonces may be random numbers, serial numbers or timestamps, but does not say much more. That is, the problem with synchronizing clocks are mentioned as a drawback to timestamps. Another comprehensive book, "Computer Security: Art and Science" [8], also provides next to nothing on nonces. A brief passage is all, with a few extra sentences for timestamps added to it. The book "Applied Cryptography" [9] is also mostly silent on the topic.

The book "Protocols for Authentication and Key Establishment" [10] does cover nonces and nonce usage. In Chapter 1.5 "Freshness" is handled. The text recognizes *Timestamps*, *Nonces (Random Challenges)* and *Counters* as ways to achieve freshness. This classification is somewhat different from the one found in the HAC or in [5], [7]. The authors also places emphasis on key freshness, which may be assured by use of nonces.

The book "Cryptography Engineering" [11] barely mentions nonces, in conjunction with Initialization Vectors (IVs), and have only a brief mention of timestamps. Here, they highlight what they call the "same-state" problem, which in effect seems to be lack of uniqueness.

C. Some Research Papers Covering Nonces

In the paper "Nonce-Based Symmetric Encryption" [12] the author discusses encryption mode-of-operations using IVs. The IVs in question are nonce based, and here the authors emphasises the uniqueness property. In the "Encode-then-encipher encryption: ..." paper [13], the general assumption is that the nonce is a counter or a random value. Collisions are problematic, and so the collision probability must be kept low.

An analysis of nonces used for authentication in a smart-card context can be found in [14]. The authors seems only to be concerned about "random" nonces, although it is not clear from the text whether they intend this to be random or pseudo-random. Still, it appears to be a trivial application where unpredictability and uniqueness seems the essential characteristics. Freshness is not directly mentioned, but it is probably assumed. Use of nonces in SIP is studied in [15]. The basis is a Diffie-Hellman exchange [16] and the use of random nonces, one for the client and one for the server. The scheme is relatively complicated, and somewhat bewilderingly, there are no other requirement on the nonces than they be "random". We assume this implies freshness and uniqueness.

The lack of preciseness and explicitness as demonstrated by the two latter papers seems to be fairly common, and informal browsing of the literature indicates that this state of affairs are the norm for many conference papers.

D. The Key Wrap Problem and Synthetic IVs

During the late 1990s, the National Institute of Standards and Technology (NIST) posed the so-called "Key Wrap" problem. The essential problem is how to develop secure and efficient cipher-based key encryption algorithms. This call caused quite a lot of research activity and then standardization. Amongst the notable papers are [17], in which the authors discusses use of the so-called *Synthetic IV (SIV)* and how to have misuse-resistant nonce-based authenticated encryption. Use of SIVs are also captured in RFC 5297 [18].

E. Keeping Time in the Face of An Intruder

Maintaining synchronized clocks in a distributed network is no small problem in itself, and it gets worse when one may expect a dishonest party. As highlighted in the Network Time Protocol (NTP) ver. 4 [19], there are serious security considerations for synchronized global clocks. Section 15 in [19] goes into considerable detail about the security implications.

F. Same-State and Time Resolution

Timestamps may suffer from resolution problems too, also known as the the same-state problem. That is, a timestamp nonce is not unique within the time resolution period. The

 ISBN: 978-1-61208-427-5

problem can be solved by combining timestamps and counters. One such scheme is presented in a paper by Mitchell [20], with a hybrid timstamp/counter based nonce. The Mitchell approach was the input to the sequence number scheme used in the UMTS Authentication and Key Agreement (AKA) protocol (Annex C in [21]). To avoid tracking by the sequence numbers, the actual sequence number is masked by an anonymity key in the UMTS AKA protocol.

III. Nonces and Formal Verification

This is just a few examples of the assumptions made by formal verification tools with regard to nonce properties.

A. BAN Logic

The research report "A Logic of Authentication" [4] did in many ways bootstrap the field of formal security protocol analysis. The logic devised in the report, commonly known as the BAN logic, is a belief logic and it is, by today's standard, both incomplete and flawed. However, it is interesting to note that the so-called "freshness formula", is essential to the logic.

1) The Freshness Formula:

$$\#(X) \tag{1}$$

> The formula X is *fresh*, that is, X has not been sent in a message at any time before the current run of the protocol. This is usually true for nonces, that is, expressions generated for the purpose of being fresh. Nonces commonly include a timestamp or a number that is used only once, such as a sequence number.

2) The Nonce-Verification Rule: The rule is concerned with freshness, and if and only if the message is fresh/recent will the receiver *believe* in the message. It is noteworthy that [4]:

> "This is the only postulate that promotes from $|\sim$ to $|\equiv$."

That is, the *nonce-verification* rule is the only rule that promotes *once said* to *believes*. This again means that the freshness property is required in BAN logic in order to successfully prove that the target protocol was successful. We observe that (pseudo-) randomness is not actually mentioned here, but it not excluded either. Timestamps and serial numbers are explicitly mentioned. Formal verification tools for verifying security protocols must necessarily have some way to capture what the nonce is to achieve. In BAN logic [4], it is clear that *freshness* is the main nonce objective.

B. Process Algebra

The book "Modelling and analysis of security protocols" [22] covers formal modelling and analysis in the context of the process algebra "Communication Sequential Processes" (CSP). There is an associated compiler, Casper, and a model checker, the "Failures-Divergence Refinement" (FDR) tool.

The authors discuss nonces and nonce properties. The Station-to-Station (STS) protocol is one of the protocols discussed, and interestingly they note that "*Note how the Diffie-Hellman terms double as nonces, so providing assurances of freshness.*" (Section 0.2 in [22]). This dual role can be problematic, unless explicitly defined and unless the properties matches both roles. Nonces is defined as (Section 0.7 in [22]):

> In the context of security protocols a nonce can informally be taken to be a fresh, random value. It is created as required in a way that is supposed to guarantee that it is unpredictable and unique. Giving precise formal meaning to terms like unpredictable, fresh and unique is itself rather subtle and we will see later how this can be done in a number of frameworks, including our own.

The authors differentiates between nonces, where randomness is involved, and timestamps. They also notes that "*If we are using time-stamps to help maintain the security of the protocols we need to bear in mind ways that an intruder might try to subvert the mechanisms.*". There is no mentioning of counters or serial numbers. That is, the authors briefly mention "identifiers for runs" and this may indeed be a counter.

As a modelling strategy, a nonce must be strictly unique in CSP. This in effect precludes the use of timestamps, and it explains why the authors distinguishes between nonces and timestamps. Also, a nonce can only be used once. This shouldn't be a surprise, but it is quite common to allow a nonce to remain within a protocol sequence as a way of binding the message sequence together (providing association). But, as noted in `Principle 6` in [5], this is a task that perhaps is best solved by other means. There are also schemes to differentiate Initiator (I) and Responder (R) nonces, although this has no real bearing on the nonce properties. Nonces are defined by the $Nonce$ constructor, where $Nonce.n$ defines n as a nonce. It is possible to capture the notion that n should be secret, but this is not a characteristic of n per se.

C. The AVISPA and AVANTSSAR Toolsets

The AVISPA project was funded by the European Union during the early/mid 2000. The acronym AVISPA stands for Automated Validation of Internet Security Protocols and Applications. The project goals was to develop formal modeling techniques for the analysis of security protocols. The formal verification tools rely on an abstract notion of the nonce, where freshness and uniqueness are the main properties [23].

The follow-up AVANTSSAR project provided considerable improvements to the AVISPA formal modelling tools, in particular the somewhat awkward High Level Protocol Specification Language (HLPSL) was substituted with the AVANTSSAR Specification Language (ASLan++) [24], which more closely resembles the familiar Alice-Bob notation. However, the semantics for the nonce is the same, and symptomatically, the way to initialize a nonce in ASlan++ is by assignment with a value from the pre-defined `fresh()` function.

While freshness may appear to be the only aspect captured here, we note that secrecy may additionally be defined as a goal for the nonce. This would captured as a protocol goal rather than as an intrinsic aspect of the nonce.

IV. Nonce Attributes

In the following section we try to capture the attributes we may require of a nonce. It is a synthesis of requirements, explicit and implicit, found in the literature.

A. Base Attributes

We need to define the attributes that a nonce may or may not have in order to properly classify the nonce types. The definitions are given with respect to a nonce of a given bit field size. The bit field size is assumed to be large enough to exceeds any practical computational means for exhaustive guessing or testing with respect to any given property.

 ISBN: 978-1-61208-427-5

B. Uniqueness Properties

We have two uniqueness variants.

Attribute 1. ***Uniqueness***

Uniqueness is the property that that the value of the nonce will never be repeated.

This is another way of expressing the once-only property. It also implies a collision-free property. The uniqueness may be expressed within a given context, and if so, it is prudent practice to make this constraint explicit. The uniqueness property effectively precludes the use of a random function for assigning the nonce value. Pseudo-random numbers will generally satisfy this property, as will sequence numbers. Uniqueness, if it can be demonstrated, is one way to ensure the never-used-before aspect of freshness.

Attribute 2. ***Statistical Uniqueness***

A statistically unique nonce shall be computationally indistinguishable from a unique nonce.

The probability of a collision for a statistically unique nonce must be exceedingly low. It must be computationally infeasible to distinguish between a unique nonce and a statistically unique nonce. If nonces are truly randomly generated, then one *may* experience collisions. However, with uniformly distributed random draws with a very large outcome space, collision will be exceedingly infrequent.

C. Freshness and Recentness

Uniqueness, or statistical uniqueness, does not quite capture timeliness or recentness. This is a problem since in many protocols it is important to demonstrate that all principals, including certificate authorities and authentication centers, are online during the protocol run. Ideally, we want to capture recentness directly in the nonce, but this harder than one might initially envision. Proving that all principal parties are online may therefore best be carried out by application of nonces, rather than being an inherent property of "fresh" nonces. We therefore exclude recentness as an inherent property of freshness.

Attribute 3. ***Freshness***

Freshness is the attribute that a certain nonce value has never been used before in the given context.

Freshness is an essential property and it is likewise essential that the freshness is verifiable by the receiver. If freshness verification, for the receiver, is not implicit in the scheme, we prefer to label it as *weak freshness*. Then, to attain *verifiable freshness* may require additional steps in the protocol itself. This is acceptable, provided that the verification is indeed carried out.

D. Unpredictable Nonces

Uniqueness and freshness are fundamental properties, but there are cases where they are insufficient or unnecessary. For instance, unpredictability is not intrinsic to uniqueness or freshness. Nonces that only require uniqueness and freshness may be produced by a sequential number scheme, but they will certainly not be unpredictable.

For some cases, one must explicitly require that the nonce is unpredictable. That is, the nonce must appear to be randomly drawn. We generally assume, but do not require, that the randomness is pseudo-randomness. The pseudo-random number generator ($prng()$) function must have uniform distribution over the whole outcome space and it must be computationally infeasible to guess the next value even when given very long series of previous values. The quality of the $prng()$ function and security associated with the initialization of the function is essential. In [25], we find many examples of security being compromised due to what the authors call "bad randomness". We define unpredictable here as compliant with our statistical uniqueness requirement, and hence it is really a "statistically unpredictable" attribute.

Attribute 4. ***Unpredictable***

This is the property that predicting the value of the next nonce must be computationally infeasible. This must hold even if the adversary have observed a large number of previous nonces and the associated contexts.

E. Predictable Nonces

Freshness can be assured without randomness, and timestamps and counters are obvious solutions. Encoding-wise, a timestamp may be seen as a special case of a counter. The counters must be monotonically increasing, although discrete continuity seems not to be required. This allows for timestamps to be interpreted as counters. It is also possible to abandon the requirement for monotonicity, but then obviously one can no longer guarantee sequential order. Schemes exits where a window mechanism is used (See SEQ use in UMTS [21]), but this opens up to the risk of replays and one looses the possibility to guarantee that the principals are all online during the transaction. Use of window mechanisms may be permissible under some circumstance, but we generally argue against it.

Counter wrap-around is a potential hazard for sequentially ordered nonces. We mention this explicitly since it is not uncommon to use protocol frame numbers and similar as "nonce-like" inputs, and these counters will generally be allowed to wrap-around. Special care must be taken for cases where this may occur.

Attribute 5. ***Sequentially ordered***

This is the property that the nonce is a counter with monotonically increasing values.

Since sequential order alone cannot provide timeliness, we have added timeliness as a separate attribute.

Attribute 6. ***Timeliness***

Timeliness is the property that the nonce is encoded as a unique timestamp. The clock source(s) must be trusted and clock synchronization must be protected.

We note that the collision-free property is satisfied for the predictable nonces. However, be warned that counter wrap-around and period resolution problems may re-introduce the same-state problem, and thereby the potential for collisions.

F. Secrecy and Authenticity

Privacy is a concern and sequential nonces may be used for tracking purposes. To avoid this one may require the nonce to be encrypted (data confidentiality).

 ISBN: 978-1-61208-427-5

Attribute 7. ***Secrecy***

Secrecy is defined to be data confidentiality for the nonce.

In some protocols, there is also a need for verified nonces. We believe that this should be explicitly captured, and hence that a integrity/authenticy attribute is needed.

Attribute 8. ***Authenticity***

Authenticity is defined to be data integrity for the nonce.

Authenticity is here equivalent to the *once said* property of BAN logic.

G. Basic Nonce Types

We now define a few basic nonce types. These will be defined in the context of the base attributes. We have qualified the freshness attribute with *weak* and *verifiable*, depending on whether the responder is directly able to verify the freshness.

Definition 1. *Random Nonce*

- *Statistical Uniqueness*
- *Weak Freshness*
- *Unpredictable*

Definition 2. *Pseudo-random Nonce*

- *(Guaranteed) Uniqueness*
- *Weak Freshness*
- *Unpredictable*

Definition 3. *Sequential Nonce*

- *(Guaranteed) Uniqueness*
- *Sequentially ordered*
- *Freshness*

Definition 4. *Timestamp Nonce*

- *Uniqueness (with resolution constraints)*
- *Verifiable Freshness*
- *Sequentially ordered*
- *Timeliness*

H. Complex Nonce Types

A *Timestamp Nonce* may suffer from the same-state problem. It may be mitigated with a high-resolution timestamp, but this introduces its own problems. For strict real-time environments this may be irrelevant. Alternatively, one may relax the timeliness guarantee to be within a slightly longer period, while maintaining the requirement for strict sequential order. That is, one may augment a timestamp nonce with a serial number to alleviate the same-state problem while avoiding a high-resolution timestamp.

Definition 5. *Augmented Timestamp Nonce*

- *(Guaranteed) Uniqueness*
- *Verifiable Freshness*
- *Sequentially ordered*
- *(relaxed) Timeliness*

We define a *Encoded Nonce* to be a sequential, timestamp or augmented timestamp nonce that has been "encoded". With encoded we here mean that the target nonce has been transformed by some cryptographic function such that the output will appear to be random and unpredictable to an external observer. However, to the initiator/responder, it is possible to reverse the encoding and retrieve the original sequential nonce. One such scheme is mentioned in Section V-A. Note that such schemes may be convoluted, and that decoding may potentially only be possible in the later phases of the protocol. One may also envisage encoded nonces that are based on random nonces, but we have not found reason to investigate this further.

Definition 6. *Encoded Nonce*

- *(Guaranteed) Uniqueness*
- *Freshness*
- *Sequentially ordered*
- *Unpredictable*
- *Optionally: Timeliness*

I. Nonce Qualifiers

The above nonce types, which should be well in line with suggestions in the literature, may optionally be augmented by what we call the nonce qualifiers. The use of qualifiers should be based on requirements on the nonce in the protocol. The previously defined basic nonce types may then be qualified as:

- **Secret** – possessing the secrecy attribute
- **Authentic** – possessing the authenticity attribute

Note that an *Encoded Nonce* does not necessarily achieve secrecy, and cannot be used in this way unless the required properties are explicitly confirmed.

V. Some Nonce-based Information Elements

A. Initialization Vector

An IV is a value used as the initial input to a cipher function mode-of-operation. For instance, in *cipher block chaining* (CBC) mode, one uses the ciphertext output of the previous stage as input to be mixed (XOR'ed) with the plaintext block. That is, the IV is the input to the initial stage of a CBC chain, where there is no previous output available.

In Section 4.3 in [11], the authors describes IV use in CBC and how to choose the IV. Specifically, they argue that one may use a "Nonce-Generated IV". The specific scheme proposed involves using a message number, which needs to be unique within its context, and then encrypting the message number to produce the nonce. The nonce is then used as the IV. However, message numbers are generally predictable and visible to the intruder, and so one must be careful not to allow the intruder to use the message number as a source for known plaintext attacks. We note that a counter/serial number that is encrypted and then used as an IV, would be a kin to a *Encoded Nonce*.

B. Random Challenge

By definition a normal random challenge information element may be a *Pseudo-random Nonce*. Potentially, it may also be *Random Nonce*, but this would be the exception. Depending on the protocol, it may be desirable to have an *Authentic Pseudo-random Nonce*.

 ISBN: 978-1-61208-427-5

C. Key Agreement

It is quite common to have a key derivation function, $kdf(\cdot)$, that accepts nonces as inputs. One example is the $f4()$ function in UMTS [26].

$$f4_K(RAND) \rightarrow IK \tag{2}$$

In (2), the random challenge is accepted as input and the integrity key, IK, is the output. The key, K, is the permanent pre-shared subscriber symmetric-key credential. In this case the $RAND$ is equivalent to an *Pseudo-random Nonce*.

D. Pseudonymous Identifiers

Privacy is of growing concern, and *identity privacy* is a pronounced concern for authentication protocols, where corroboration of a claimed identity (or identifier) is a primary goal (Chapter 4 in [27]).

With symmetric-key schemes one has no option but to present the claimed identifer in plaintext form. This invariably exposes the identifier and identity privacy is lost. For a mobile subscriber, *location privacy* is simultaneously lost. One way to solve the problem is to use asymmetric cryptography and hide the presented identifer so that only the intended recipient is able to decrypt the message. It is quite common with such protocols to also forward (or agree on) a temporary identifier, to be used subsequently [21], [28]. This temporary identifier must then be free of any apparent association with the primary identifier. That is, to any external observer there must not appear to be any correlation between the primary identifier and the temporary identifier. This, of course, can only be done if the temporary identifier appear to be random with respect to the primary identifier.

The temporary identifier must be unique, or at least statistically unique, within the given context. A *Pseudo-random Nonce* will nicely fit this description.

VI. An Example of Nonce Usage

We have chosen to use the UMTS AKA protocol as an example protocol. It should be a relevant example, as it is both a fairly simple protocol and it is a widely used protocol.

A. Case Study: The UMTS AKA Protocol

The UMTS AKA protocol includes a random challenge and a sequence number scheme. The random challenge is clearly a nonce and the sequence number, the SQN field, fits the *number used once* requirement. The primary reference for the UMTS AKA protocol are the 3GPP technical specifications TS 33.102 [21] and TS 33.105 [26]. See also [29] for an overview.

The UMTS AKA protocol is a 3-way protocol with the following principal entities:

- **User Equipment (UE)** – represents the subscriber.
- **Visited Network (VN)** – the local access network.
- **Home Network (HN)** – UE subscription point.

A central component of the protocol is the Authentication Vector (AV):

$$AV = \{RAND, XRES, CK, IK, AUTN\} \tag{3}$$

The $RAND$ is the pseudo-random challenge (128 bit), $XRES$ is the expected response (64 bit), CK and IK are the session keys (128 bit each) and the $AUTN$ is the authentication token (128 bit) compound IE. $XRES$ and RES are identical, but that the former is the expected response while the latter is the actual response (as computed by the USIM). The Authentication Token ($AUTN$) is defined as:

$$AUTN = \{SQN \oplus AK, AMF, MAC{-}A\} \tag{4}$$

Here AK is an anonymity key, AMF is the authentication management field and $MAC{-}A$ is the cryptographic check sum for the challenge.

The Protocol:

1) VN $\rightarrow$ HN: SendAuthInfo-req($IMSI$)
2) HN: Generate AV
3) HN $\rightarrow$ VN: SendAuthInfo-resp(AV)
4) VN $\rightarrow$ UE: AuthReq($RAND, AUTN$)
5) UE: Compute $MAC{-}A, RES, CK/IK$ and AK
6) UE: Verify validity of challenge
7) UE: Verify timeliness of challenge
8) UE $\rightarrow$ VN: AuthResp(RES)
9) VN: Verify that $RES = XRES$

The challenge is verified by checking $MAC{-}A$. The timeliness is verified based on the sequence number. The SEQ may be seen as a *Secret Sequential nonce*. The secrecy is provided by the used of the anonymity key (AK).

The $RAND$, in conjunction with the associated $AUTN$, may be seen as an *Authentic Pseudo-random Nonce*.

VII. Discussion and Analysis

A. Being Explicit

We note that Principle #6 in [5], which said "*Be clear what properties you are assuming about nonces.*", tend to be ignored. That is, the requirements on the nonce may have been clear to the designer(s) of the protocol, but the requirements must be made explicit and well documented. This will not only remind the designers about what assumptions there are on the properties of the nonce, but it will also be essential in order to carry out formal verification of the protocol. And, clearly, it is vital for the implementors that all assumptions are explicit.

B. Multiple Usages

As exemplified by the UMTS AKA protocol, the random challenge nonce is used for different purposes simultaneously, including entity authentication and key derivation. This may be permissible, but then certainly these nonce requirements must be explicit. It is also not clear that it is advisable to use a nonce for different purpose, even if the nonce properties does match the nonce usage well. That is, this may be detrimental to security in much the same manner as using a shared secret key for different algorithms may lead to added vulnerability.

C. Formal Verification and Type Systems

In [30], [31] there are deep and broad background coverage of type theory and type systems in computer languages. The last part is quite relevant, as formal modelling of security protocols is captured in a modelling language. To enhance a modelling language to capture more aspects of nonces and perhaps even to have a set of different nonce types would clearly be useful. The properties can then at least be checked for consistency and one may even prove that certain properties are adhered to (or not).

 ISBN: 978-1-61208-427-5

However, as noted by Gollmann in [32], proving a protocol correct may not be the most important aspect (Gollmann even declares this to be a non-goal). In fact, the most important aspect may indeed be to adhere to `Principle 6` [5]. This would make it easier to define exactly what the protocol is to achieve in the first place, and it would make it easier to construct the protocol. In [33] this is recognized, and the authors does see formal verification tools in the context as a design aid. Correctness is one thing, but improved clarity and less ambiguity in the description may also lead to more reliable and robust implementations.

We agree with this view and would welcome further research in capturing nonce properties in type systems for use with formal verification tools. This would represent one step in the right direction for designing better and more secure protocols.

VIII. Summary and Concluding Remarks

In this paper, we have made a brief survey of how nonces are discussed in the literature. The survey is by no means exhaustive, but then even our selected sources demonstrates quite well that description of nonce properties in cryptographic protocols often are vague or even missing. And, certainly, the descriptions are almost invariably incomplete.

We have not strived for completeness, or even rigorousness, but we have attempted to further the field by providing improved nonce attribute characteristics and better and more useful nonce definition. However, more work is needed here and we intend to extend our study of nonce usage by investigating more cryptographic protocols. We will also continue our work with type systems for nonces and investigate ways to verify the properties.

Ultimately, we believe greater awareness about nonce properties should lead to less ambiguity and thereby better designs for security protocols. Clarity in what a security protocol should achieve is obviously essential and a goal in itself. Protocol designs may certainly benefit from being captured in a formal languages, not only because it may allow for formal verification of selected properties, but also because increased awareness and preciseness ultimately may lead to better implementation too. In the end then, we may have hope for better, more reliable and robust security protocols.

References

[1] N. P. Smart, V. Rijmen, M. Stam, B. Warinschi, and G. Watson, "Study on cryptographic protocols," ENISA, Report TP-06-14-085-EN-N, 11 2014.

[2] N. P. Smart *et al.*, "Algorithms, key sizes and parameters report - 2014," ENISA, Report TP-05-14-084-EN-N, 11 2014.

[3] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography; Fifth Printing (August 2001)*. CRC press, 2001.

[4] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," DEC System Research Center, Research Report 39, 2 1990.

[5] M. Abadi and R. Needham, "Prudent engineering practice for cryptographic protocols," DEC System Research Center, Research Report 125, 6 1994.

[6] J. Clark and J. Jacob, "A survey of authentication protocol literature: Version 1.0," 1997.

[7] R. Anderson, *Security engineering*. John Wiley & Sons, 2008.

[8] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley, 2003.

[9] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. john wiley & sons, 1996.

[10] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, 1st ed. Springer Science & Business Media, 2003.

[11] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications: Design Principles and Practical Applications*. John Wiley & Sons, 2011.

[12] P. Rogaway, "Nonce-based symmetric encryption," in *Fast Software Encryption*. Springer, 2004, pp. 348–358.

[13] M. Bellare and P. Rogaway, "Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography," in *Advances in Cryptology (ASIACRYPT 2000)*. Springer, 2000, pp. 317–330.

[14] N. Junghyun, K. Seungjoo, P. Sangjoon, and W. Dongho, "Security analysis of a nonce-based user authentication scheme using smart cards," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, no. 1, pp. 299–302, 2007.

[15] J. L. Tsai, "Efficient nonce-based authentication scheme for session initiation protocol," *International Journal of Network Security*, vol. 9, no. 1, pp. 12–16, 2009.

[16] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[17] P. Rogaway and T. Shrimpton, "A provable-security treatment of the key-wrap problem," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Springer Berlin Heidelberg, 2006, vol. 4004, pp. 373–390.

[18] D. Harkins, "Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)," IETF, RFC 5297, 10 2008. [Online]. Available: https://tools.ietf.org/html/rfc5297

[19] D. Mills, J. Martin, J. burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," IETF, RFC 5905, 06 2010. [Online]. Available: https://tools.ietf.org/html/rfc5905

[20] C. J. Mitchell, "Making serial number based authentication robust against loss of state," *ACM SIGOPS Operating Systems Review*, vol. 34, no. 3, pp. 56–59, 2000.

[21] 3GPP, TS 33.102, "3G Security; Security architecture," 3GPP, France, TS 33.102 (3G), 2014.

[22] P. Ryan, S. A. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe, *The Modelling and Analysis of Security Protocols: the CSP Approach*. Addison-Wesley Professional, 2001.

[23] AVISPA project, "Automated Validation of Internet Security Protocols and Applications (AVISPA); AVISPA v1.1 User Manual," AVISPA IST-2001-39252, Tech. Rep., 06 2006.

[24] AVANTSSAR project, "Aslan++ specification and tutorial," FP7-ICT-2007-1, Deliverable 2.3 (update), 01 2008.

[25] B. Schneier, M. Fredrikson, T. Kohno, and T. Ristenpart, "Surreptitiously weakening cryptographic systems," Cryptology ePrint Archive, Report 2015/097, 2015.

[26] 3GPP, TS 33.105, "3G Security; Cryptographic algorithm requirements," 3GPP, France, TS 33.105 (3G), 2014.

[27] G. Danezis *et al.*, "Privacy and data protection by design from policy to engineering," ENISA, Report TP-05-14-111-EN-N, 12 2014.

[28] G. M. Køien, "Privacy enhanced cellular access security," in *Proceedings of the 4th ACM workshop on Wireless Security*. ACM, 2005, pp. 57–66.

[29] ——, "An introduction to access security in UMTS," *Wireless Communications, IEEE*, vol. 11, no. 1, pp. 8–18, Feb 2004.

[30] B. C. Pierce, *Types and programming languages*. MIT press, 2002.

[31] L. Cardelli, *Computer Science Handbook*, 2nd ed. CRC press, 6 2004, ch. Type Systems.

[32] D. Gollmann, "Analysing security protocols," in *Formal Aspects of Security*. Springer, 2003, pp. 71–80.

[33] J. A. Clark and J. L. Jacob, "Protocols are programs too: the meta-heuristic search for security protocols," *Information and Software Technology*, vol. 43, no. 14, pp. 891–904, 2001.

 ISBN: 978-1-61208-427-5

A Model for Conducting Security Assessment within an Organisation

Nor Fatimah Awang
Faculty of Defence Science and Technology
National Defence University of Malaysia
Kuala Lumpur, Malaysia
e-mail: norfatimah@upnm.edu.my

Azizah Abd Manaf
Advanced Informatics School (UTM AIS)
UTM International Campus
Kuala Lumpur, Malaysia
e-mail: azizaham.kl@utm.my

***Abstract*—Security Assessment is widely used to audit security protection of web applications. However, it is often performed by outside security experts or third parties appointed by a company. The problem appears when the assessment involves highly confidential areas which might impact the company's data privacy where important information may be accessed and revealed by the third party. Even though the company and third party might have signed a non-disclosure agreement, it is still considered a high risk since confidential information on infrastructure and architecture are already exposed. It is important to keep the confidential information within the project team members to protect the data used by the system. Therefore, this paper proposes a model to conduct internal security assessment to ensure all organisational assets are protected and secured. The main objective of this paper is to discuss the activities and processes involved in conducting the security assessment.**

Keywords-Web application; vulnerability; security testing; security assessment; penetration testing.

I. INTRODUCTION

Today, more than one billion people worldwide use the Internet in their daily routine for a variety of reasons, such as communicating with others, conducting research, shopping, banking and electronic commerce [1]. Due to the high usage of the Internet in today's highly competitive world, more organisations are relying solely on web-based applications and the Internet to change their daily manual activities to online-based activities. Most of the organisations have shifted to the Internet to make more profits and at the same time to increase the efficiency of their activities such as customer support services, data transactions and quality of information supply [2]. From businesses, industries, governments to non-profit organisations, the Internet has simplified a lot of business processes and activities. The growth of internet applications gave a high impact and created business opportunities to the organisations. However, the Internet has also brought unintended consequences, such as criminal activities, spamming, credit card frauds, online fraud, theft of sensitive information, phishing and other related cyber-crimes [3][4]. According to surveys conducted by the security firm McAfee and the Center for Strategic and International Studies, millions of dollars have been lost due to cyber-crime attacks [3]. In fact, Symantec Group reported, attacks against web applications have increased in 2010 by 93% compared to 2009. Another report showed that, almost 150,000 new sites are registered per day on the internet, which has the potential to introduce around two billion serious vulnerabilities [5].

There are numerous researches that focused on the issues of web application security and vulnerability. Many of the studies provide models, methodologies and technologies to enhance the security in web applications. One important step to ensure web application security is to conduct security assessment periodically. Security assessment is a process to search for potential loopholes or vulnerabilities contained in a system. Through the security assessment, an organisation can then assure that systems and applications are operating effectively in providing appropriate product or service confidentiality, integrity and availability [6]. The assessment is important to make sure all systems are secure and all vulnerabilities are discovered before any system is being deployed [7][8]. Some companies choose to use consultants or outsource security assessments to third-parties. Outsourcing security assessment is mandatory in security audit for banking and online business industries, therefore a software industry for any related business can just concentrate on developing their system and let the third-party evaluate their product before releasing it to the market. However, according to a study conducted by Corwill and Nasimmbeni, there are some security issues involved when using external party to conduct an assessment [10][11]. Even though a non-disclosure agreement has been signed by both parties to prevent them from divulging information, it is still considered a high risk as the third-party already has the confidential infrastructure and architecture information. It is therefore important to keep the internal information within project members to protect the confidential data used by the system.

This paper discusses a model for conducting security assessment and detecting vulnerabilities that exist in web applications. Security assessment is a process to find potential security loopholes or vulnerabilities in target systems. Using this model, many organisations will have the opportunity to perform security assessment internally without having to outsource it to third-party security experts.

 ISBN: 978-1-61208-427-5

The structure of this paper is as follows. Section II briefly describes the background of web application architecture and discusses related techniques which are commonly used in detecting vulnerabilities in web applications. Next, Section III discusses in detail the proposed model. Section IV discusses the results and finally, Section V presents the conclusion.

II. BACKGROUND AND RELATED WORK

A. Web Application Architecture

Since a web application runs in the dynamic and distributed environment that is different from the traditional programmes, hence more vulnerability exists. This section gives some explanations on the architecture of web applications and several common vulnerabilities which exist in web applications. In general, a web application has three tier constructions as shown in Figure 1 [12][13]. Figure 1 describes the architecture of a web application. The architecture of a web application consists of web browser, web server, web application and database server. In Tier 1, web server receives input and interacts with clients through web browser by using http or http protocol. The web applications are developed using different programming language such as Active Server Page (ASP), Common Gateway Interface (CGI), Ruby or Java in Tier 2. Generally, the web server will manage the page requested from the web client by sending the request to the application server and the application server constructs codes dynamically and passed the codes back to the web server. The flow of data amongst the tiers gives rise to input validation problem for the web application server; it must check and/or modify incoming the input before processing it further or incorporating the input into output that it passes to other tiers to execute. Failure to check or sanitise the input appropriately can compromise the web application's security [14]. Similarly, Tier 3 is responsible for the access of authenticated users and rejection of malicious users from the database.

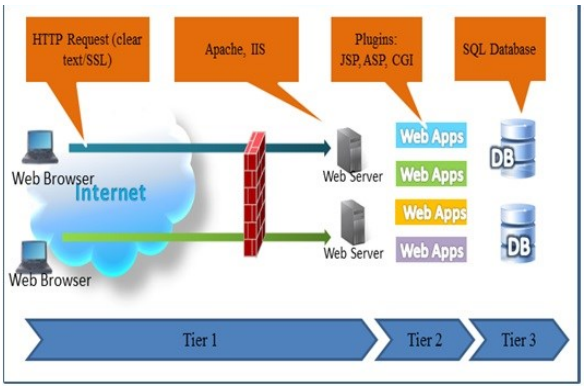


Figure 1. Web Application Architecture

B. Web Application Vulnerabilities

In this paper, the definition of web application vulnerabilities follows the definition from the Open Web Application Security Project (OWASP) [18], which defines vulnerability as a hole or a weakness in the application, which can be a design flaw or an implementation bug that allows an attacker to cause harm to the application.

There are a few web application vulnerability databases available on the Internet, e.g., OWASP Top 10 Web Application Vulnerability [18], SANS Top 20 2007 Security Risks, and WASC Threat Classification [15]. These databases classify and identify all known web application vulnerabilities and attacks, and they are continuously updated and maintained. The public security knowledge databases are very useful to testers for self-education and used as test references. With lots of vulnerabilities appearing in web applications, it is more difficult for system or network administrators to protect core assets such as personal information, confidential data and customer credit card numbers. The most common and popular vulnerabilities exploited by attackers are SQL injection and cross site scripting [16]. These are due to improper sanitisation in input validation fields. The researcher in [17] highlighted some potential vulnerabilities that will help security tester or assessor to understand possible vulnerabilities in login page that would be useful for security assessment.

The model of this study aims to identify potential vulnerabilities which exist in web applications. The goals of testers are to mimic the possible techniques commonly used by the attacker to attack the system, identify possible vulnerability based on the functionality, identify test cases to the system and find out how to exploit these attacks to improve the web application security. The testers can obtain information that help them to understand what are the function and vulnerability that are commonly used by the attacker to exploit the system by analysing the intentions of functionality and vulnerability. This paper extends the results presented in [8] and [9].

C. Techniques to Detect Vulnerability

There are many techniques for detecting vulnerabilities during the process of software development life cycle such as static code analysis, dynamic analysis and security assessment, and penetration testing. Static Analysis consists of analysis on the application source code [19]-[21]. It is performed on the source code without executing the application. This can be done manually or by using code analysis tools such as FORTIFY, Ounce or Pixy [22]. Reports are generated and presented to the developer team after reviewing the source code. Generally, it helps to catch implementation structural bugs early and it is important to know that static analysis cannot solve all security problems. There are different tools available now for this kind of test but it is not easy to find mature and well tested tools to discover all the security defects in an application. The problem is that code analysis may be difficult and may not find all security flaws because of the complexity of the code [23].

Dynamic Analysis, also known as Dynamic Testing is used to test a program by executing it in real-time [24]. Dynamic Analysis test will communicate with a web

 ISBN: 978-1-61208-427-5

application through the web browser in order to identify potential security vulnerabilities and architectural weaknesses in the web application. The objective is to find security errors in the web application while it is running. This technique can be performed either manually or by using automated tools [25]. Automated tool provides an automatic way to search for vulnerabilities by avoiding the repetitive and tedious task of doing hundreds or even thousands of tests manually for each vulnerability type [26].

III. THE PROPOSED MODEL

This section describes the phases and activities of the proposed model as shown in Figure 2. The three main phases are Data Gathering, Attacks, and Reporting. Each phase comprises of several major activities together with their flows and stages.

A. Data Gathering – Phase 1

This is the first stage in the model. There are six major activities involved in this stage. The first three activities are basically planning focused activities. In this phase, there are some items that should be highlighted and prepared such as identifying which target system that should be tested to detect vulnerability, and what type of potential threat or vulnerability that commonly exists in web applications. Additionally, questions such as how long the testing will be carried out, which methodology will be used and what restrictions or limitations need to be applied must be tackled. The test plan should also outline the tools needed to conduct the tests, as well as exploring opportunities for automated testing. Next is to find other test planning criteria as shown in Table I. The tools used for the assessment are combinations of both commercial and open source software. At least two different tools are used to perform the test to ensure accuracy of the result. Table II lists the tools used during this assessment.

The other three activities, as discussed below, are more hands-on and mostly based on the first three activities in data gathering and findings.

Scanning - This phase is more on mapping of the potential vulnerabilities detected by scanners with main system components. This activity uses the vulnerability scanner to scan the services in identifying potential loop holes and vulnerabilities in web applications.

Discovery Scanning Analysis - In this activity, results produced by different tools are compiled for further analysis.

Risk Rating - In this activity, discovery analysis findings are used as the main source and subject in risk rating. The risk rating outcomes or results are more specific to the assessed system. The findings are then categorised in Section IV into three risk levels such as high, medium and low in order to indicate the level of severity. The severity levels are based on the guidelines from OWASP and recommendation from tools. This rating is used throughout this assessment to provide common understanding of the risk.

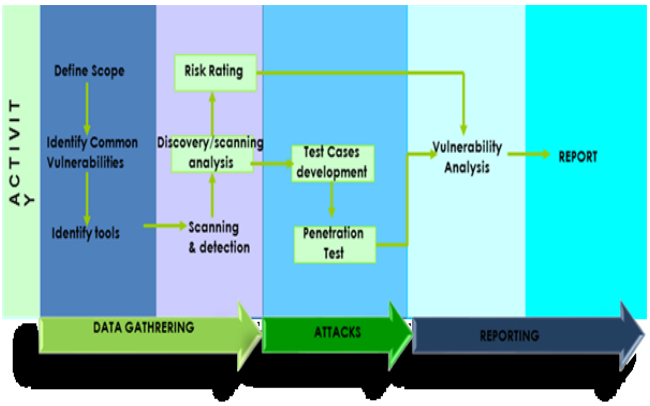


Figure 2. Model for detecting vulnerability in web applications

TABLE I. TEST PLANNING CRITERIA

Criteria	Planning Detail
No. of Security Tester and Qualification	To get the number of certified security tester and the tester unified qualification.
Type of Tools	To see if they use open source tools available on the net or commercial tools.
Number of Server	How many servers will be involved in this assessment?
Test Time Frame	How long is the duration for this assessment?

TABLE II. LIST OF TOOLS

Tools	Testing Activities
Zenmap	To get banner grabbing for server
Nessus Nexpose Burp Suite Free Edition Acunetix Web Vulnerability Scanner	During Scanning phase
Test Case generator Attack Generator	During Attack phase

B. Attacks – Phase 2

This is the second stage in the model. As the name suggests, it is responsible for performing the attacks on the system. The attacks are performed on the vulnerabilities that have been discovered during the data gathering phase. The attack phase is executed in a cascaded manner where every successful attack leads to obtaining more privileges and system information. There are two major activities involved in this phase, which are test cases development and penetration testing.

Test Cases - Structured test cases are developed based on the OWASP testing guidelines [17]. In this study, information on existing known vulnerabilities are collected and analysed to generate attack test cases. In this phase, a tool called Test Case Generator was developed to generate attack test cases. There were 1600 test cases generated to perform SQL Injection in the vulnerable web applications. Table III presents some samples of test cases which were

 ISBN: 978-1-61208-427-5

generated by the Test Case Generator. These test cases were used to perform the attack during the penetration testing phase.

Penetration Testing – Usually, in this stage penetration testing is manually performed by a security expert to confirm the vulnerabilities detected during the scanning phase (to check false positive of the vulnerabilities). In the study model, the attack generator tool which was developed is used to automatically perform penetration testing. The attack test cases generated from the previous stage were utilised to inject and detect vulnerabilities. The fundamental objective of this Section is the design of the model that covers all steps to automate the injection attack process. This tool injects abnormal input to input parameter and discovers unexpected defects and vulnerabilities. The Attack Generator starts processing a set of target URL and target parameter. Some manual works are still required before automating the attack generator process. A Tester is needed to identify the target URL and target parameter. The test cases generated in the previous stage will be used as an input in this stage. In order to extract HTTP response and injecting them to the target system automatically, the model was developed using Apache HTTP Client API. The Attack Generator component uses input.xml (Figure 3) file to attack the target system by using POST or GET method, and it also identifies which parameter is chosen to inject the test cases. The target URL, HTTP method and parameter are chosen by the researchers.

TABLE III. TYPES OF TEST CASES

Type of Vulnerability	Test Cases
SQL Injection	' or 1=1-- " or 1=1-- ' or 1=1 /* or 1=1-- ' or 'a'='a " or "a"="a %27+OR+%277659%27%3D%277659 %22+or+isnull%281%2F0%29+%2F* a' ORDER BY 1;#
Cross Site Scripting	<script>alert("TEST");</script> <script>alert("HELLO");</script> <SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>

C. Reporting - Phase 3

This final stage in the model concludes the assessment from the combination of the two main phases – data gathering and attacks. Vulnerability analysis result is based on the results of two activities from these two different phases. Mapping the risk rating conducted in Phase 1 and validation of penetration testing in Phase 2, are the major sources of vulnerability analysis. Once completed, the report will provide the identification of all security vulnerabilities found. Each finding will be assigned a risk rating based on certain criteria, together with remediation recommendations to resolve the vulnerability. This phase analyses all the vulnerabilities based on http response collected after the injection of attacks to the target application. The results are then categorised into three classes as shown in Table IV.

```
<url>http://WackoPicko/users/login.php
</url>
    <method>post</method>
    <parameters-group>
            <parameters>
            <param>username</param>
    <file> input1.txt</file>
            </parameters>
            <parameters>
            <param>password</param>
            </parameters>
    </parameters-group>
```

Figure 3. Sample of input.xml

TABLE IV. CLASSES OF RESULT

Example of attack string	Example of HTTP Response	Classes of result
‘	You have an error in your SQL syntax	SQL Error
“	The username/password combination you have entered is invalid	No error
' OR '1'='1	ID: ' OR '1'='1 First name: admin Surname: admin ID: ' OR '1'='1 First name: Gordon Surname: Brown	Bypass Application

IV. RESULTS AND DISCUSSION

This Section presents the results of tests carried out to verify the model of study. Three vulnerable websites were chosen for this experiment; WackoPicko is an online photo sharing website that allows users to upload, comment and purchase pictures, while Peruggia is a website which is similar to WackoPicko. The third website is Damn Vulnerable Web Application (DVWA), normally used as an aid for security professionals to test their skills and tools in a legal environment, and help web developers understand better the processes of securing web applications. All these websites are designed with a number of vulnerabilities, such as cross-site scripting and SQL injection. Usually, the vulnerable websites were selected by researchers to test, investigate and verify their methods or approaches [27][28]. This experiment focuses only on SQL injection vulnerability. Our model was deployed by setting up the Eclipse development environment with Java Program. The Apache HTTP Client API library was installed in the machine to extract HTTP header from response pages. In the test case generation phase, 1600 attack test injections were generated for SQL injection attack. We ran the model with attack test cases, and the results are summarised in Table V and Table VI. The response results will indicate the vulnerability if error messages and bypass authentication results appeared in the HTML document header. Based on results of the test, it could be concluded that all input forms are vulnerable to the website. Due to some constraints, Acunetix was the only

 ISBN: 978-1-61208-427-5

tool available to us at the time of writing this paper. The results of running the scanner against vulnerable web applications in the scanning phase are shown in Table V. SQL vulnerabilities were discovered in WakcoPicko and DVWA websites, but not in the Peruggia website. The function of the scanning tool is to find weaknesses in the application. The examples in Table V, when the tool inserted attack string 1 ' ", and the database error generated with SQL error in the message status, the tool will indicate that there are vulnerabilities. The tool provides only an overview of SQL error without explicitly detailing the weaknesses in the system. If seen randomly, SQL error does not give any meaning to a new tester (not an expert security tester). Thus, our model can solve problems found in Phase 2. Usually in penetration testing, the security tester will verify manually whether an attack can be executed or otherwise. Usually the attack used to verify whether the attack is successful or not (for the login form) is by using the attack string ' OR 1 = 1--.

TABLE V. DETECTION RESULT AT SCANNING PHASE

Application	Target parameter	Attack String	Result Analysis
Wacko Picko	username/ password	1'"	SQL Error
Peruggia	username/ password	None	None
DVWA	userid	1'"	SQL Error

TABLE VI. DETECTION RESULT AT ATTACK PHASE

Application	Target parameter	Attack String	Result Analysis
WackoPicko	username/ password	‘	SQL error
		1'"	SQL error
		' OR '1'='1	Bypass application
		' order by 1 #	Bypass application
		1 ORDER BY 1	No error
		"a' OR database() LIKE '%A%';#	Bypass application
Peruggia	username/ password	‘	No error
		1'"	No error
		' OR '1'='1	No error
		' order by 1 #	No error
		1 ORDER BY 1	No error
		"a' OR database() LIKE '%A%';#	Bypass application
DWVA	userid	‘	SQL error
		1'"	SQL error
		' OR '1'='1	Bypass application
		' order by 1 #	No error
		1 ORDER BY 1	Bypass application
		"a' OR database() LIKE '%A%';#	Bypass application

Table VI shows a list of attack strings which successfully bypass the application and entered the application. As seen in Table VI, WackoPicko and DVWA are the easiest to bypass the application. By simply using the simple attack string ' OR 1 = 1, one can easily enter into the application. On the other hand, Peruggia requires advance test cases to enter the application. Acunetic scanner tool could not detect any vulnerability found in the Peruggia website. The result of this study proves that the study model successfully detects vulnerability even though it cannot be detected during the scanning phase.

V. CONCLUSION

This paper aims to provide a web security assessment model for in-house self-assessment exercise which will help to identify the weaknesses and potential vulnerabilities of web applications. OWASP Top Ten vulnerabilities classification is used as the main reference or guidelines to seek security holes in the web applications and simulate hackers' actions via specific test cases to validate the real existence of vulnerabilities. The overall methodology is relatively straightforward, but the existing method was extended with newly generated test cases and analysed http response with three different classifications; SQL error, no error and bypass application. After conducting the security assessment in selected web applications, the result shows that the model has successfully detected vulnerability in the web applications even though it cannot be detected during the scanning phase. The result is then categorised based on three classifications and it was found that the class with bypass applications is with critical vulnerabilities and requires immediate action to mitigate risks. There is intention of implementing other attack types such as cross site scripting and parameter manipulation attack to replace the SQL injection in future.

ACKNOWLEDGMENT

This work was supported by the Advanced Informatics School (AIS), University Technology of Malaysia and National Defence University of Malaysia.

REFERENCES

[1] G. B. shelly and M. E. Vermaat, "Discovering Computers 2009: Living in Digital World, Complete," Cengage Learning Course Technology, 2009.

[2] A. Al-dahoud and C. Universitaria, "E-Government : Benefits , Risks and a Proposal To Assessment Including Cloud Computing and Critical Infrastructure," 2013.

[3] P. Katsumata, J. Hemenway and W. Gavins, "Cybersecurity risk management," Military Communications Conference, 2010 - MILCOM 2010 , vol. Oct. 31 2010-Nov. 3 2010, no., pp.890-895.

[4] K. Francis, B. Andoh and K. O. Bryson, "Exploring the characteristics of Internet security breaches that impact the market value of breached firms," Expert Systems with Applications, Volume 32, Issue 3, April 2007, pp. 703-725, ISSN 0957-4174.

[5] G. Jeremiah, "The State of Website Security," Security & Privacy, IEEE , vol.10 no.4, 2012, pp.91-93.

[6] J. D. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla and A. Murukan, "Improving Web Application Security: Threats and Countermeasures," Microsoft Corporation, http://msdn.microsoft.com/en-us/library/aa302420.aspx, 2003 [retrieved: July, 2015].

 ISBN: 978-1-61208-427-5

[7] A. Ahmad, S. R. Ahmad, N. F. Awang and M.Z. Ali, "Web Vulnerability Assessment: Outsource dilemmas," Electrical Engineering and Informatics (ICEEI), 2011 International Conference , vol., no., 2011, pp.1-6.

[8] P. Xiong and L. Peyton, "A model-driven penetration test framework for Web applications," Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on , vol., no., 2010, pp.173-180.

[9] Y.-W. Huang, S.-K. Huang, T.-P. Lin, and C.-H. Tsai, "Web application security assessment by fault injection and behavior monitoring," Proc. twelfth Int. Conf. World Wide Web - WWW '03, p. 148, 2003.

[10] C. Colwill, and A. Gray, "Creating an effective security risk model for outsourcing decisions," BT Technology Journal, Vol. 25 No. 1, 2007, pp. 79-87.

[11] G. Nassimbeni, M. Sartor and D. Daiana, "Security risks in service offshoring/outsourcing: an assessment model based on the Failure Mode and Effect Analysis," POMS 21st Annual Conference, Vancouver, Canada, 2010.

[12] J. G. Kim, "Injection Attack Detection Using the Removal of SQL Query Attribute Values," Information Science and Applications (ICISA), 2011 International Conference on , vol., no., April 2011, pp.1-7, doi: 10.1109/ICISA.2011.5772411

[13] Z. Su and G. Wassermann, "The Essence of Command Injection Attacks in Web Applications," In Conference Record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 2006, pp. 372-382.

[14] T. Scholte, D. Balzarotti, and E. Kirda, "Have things changed now? An empirical study on input validation vulnerabilities in web applications," Comput. Secur., vol. 31, no. 3, pp. 344–356, 2012.

[15] Trustewave, The Trustwave 2012 Global Security 2012, https://www.trustwave.com/spiderlabs, 2012. [retrieved: July, 2015].

[16] S. Zanero, L. Carettoni and M. Zanchetta, "Automatic Detection of Web Application Security Flaws," Black Hat Forum, 2005.

[17] N. F. Awang, A. A. Manaf and W. S. Zainudin, "A Survey on Conducting Vulnerability Assessment in Web-Based Application," 2014, pp. 459–471.

[18] The Open Web Application Security Project: The Ten Most Critical Web Application Security Vulnerabilities. https://www.owasp.org/index.php/Main_Page:OWASP_Top_Ten_Project, [retrieved: July, 2015].

[19] N. Jovanovic, C. Kruegel and E. Kirda, "Static analysis for detecting taint-style vulnerabilities in web applications," Journal of Computer Security, 2010, pp. 861-907.

[20] Y. Xie and A. Aiken, A, "Static detection of vulnerabilities in scripting languages," Proc. 15th USENIX Security Symposium, 2006, pp. 179-192.

[21] N. Antunes and M. Vieira, "Comparing the effectiveness of penetration testing and static code analysis on the detection of SQL injection vulnerabilities in web services," 2009 15th IEEE Pacific Rim Int. Symp. Dependable Comput. PRDC 2009, 2009, pp. 301–306.

[22] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix and W. Pugh, "Using Static Analysis to Find Bugs," IEEE Software, 2008, pp 22-29.

[23] M. Vieira, N. Antunes and H. Madeira, "Using Web Security Scanners to Detect Vulnerabilities in Web Services," IEEE/IFIP Intl Conf. on Dependable Systems and Networks, DSN 2009.

[24] R. S. Basaval, "Web application vulnerability detection using dynamic analysis with penetration testing," International Journal of Enterprise Computing and BusinessSystems, Vol 2, 2012.

[25] M. Curphey and R. Araujo, "Web Application Security Assessment Tools," IEEE Security & Privacy, Published By The IEEE Computer Society, 2006.

[26] OWASP Testing Guideline, https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents. [retrieved: July, 2015].

[27] R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," J. Brazilian Comput. Soc., vol. 20, 2014, p. 4.

[28] Z. Djuric, "A black-box testing tool for detecting SQL injection vulnerabilities," 2013 2nd Int. Conf. Informatics Appl. ICIA 2013, 2013, pp. 216–221.

 ISBN: 978-1-61208-427-5

Cloud Card Compliance Checklist

An efficient tool for securing deployment Card Solutions on the Cloud

Hassan El Alloussi, Laila Fetjah, Abdelhak Chaichaa
Department of Mathematics and Computer Science
University Hassan II, Ain Chock, Faculty of Sciences
Casablanca, Morocco
e-mail: halloussi@gmail.com, l.fetjah@fsac.ac.ma, chaichaa@fsac.ac.ma

***Abstract*—The Payment Card Industry Data Security Standard (PCI-DSS) is a standard that aims to harmonize and strengthen the protection of Card Data in the whole lifecycle. Since its introduction, it has always been an efficient tool for controlling Card data on a platform deployed internally. In addition, it has been proved that this standard is among the best one for gauging data security, because it dictates a series of scrupulous controls and how they could be implemented. However, with the coming of the Cloud, the strategies have changed and the issues in protecting Card data become more complex. In this paper, we continue our previous work by developing a checklist that will be a reference for the Cloud tenant to control the security of Card data and information on the Cloud Computing. In the next steps, we will focus on evaluating this checklist on a real Cloud environment. Afterward, we work on recommending more requirements and controls that the norm PCI-DSS could adopt to be more efficient on Cloud and later we will develop a new Self-Assessment Questionnaire as a reference for Qualified Security Assessors (QSA) to check on the environment.**

Keywords-Cloud Computing; PCI-DSS; Card Industry; PCI-SSC; Cloud Computing Alliance (CSA); Cloud Controls Matrix (CCM)

I. INTRODUCTION

As the competition puts pressure on companies to increase productivity and decrease capital investments, solutions like distributed computing, that offer scalable systems with low fees, are attractive options for management to take in consideration. However, when you are responsible for the security of the access and the network, the idea of migrating everything to an environment that is not controlled and even owned, probably makes the decision more difficult.

Therefore, many banks and card transactions companies, which are attracted to outsourcing card solution outside their premises, encounter several obstacles, mainly related to security and data governance. The client has the responsibility to know where its data are and where it is going. This concept is the basis to data security, and plays a significant role in achieving and maintaining compliance with security norms, mainly the PCI-DSS [2].

Unfortunately, most of the requirements focus on the merchant's ability to implement network access controls, data control, and insuring that the applications installed respect the security norms by periodically test their effectiveness. In addition, it may be difficult to do it and insufficient in a Cloud platform, where the infrastructure is outsourced [7].

In this paper, we continue our previous work [1] by developing a checklist that will be an efficient tool for banks and Card companies to control if the Cloud platform is ready to receive Card solutions or not. We based our contribution on two mains frameworks: Cloud Controls Matrix (CCM) [6] developed by Cloud Computing Alliance (CSA) and PCI-DSS.

In the next section, there is an explanation of the main advantages of the CCM [4] and its domains. Section III explains the choices of domains on what we focus on. Section IV details the matrix developed and the correspondent checklist for client that allow them to verify the effectiveness of the platform outsourced (we give an extract of the checklist in Table I). Section V brings a critical view to PCI-DSS standard insufficiency in Cloud computing. Finally, we draw a conclusion in Section VI.

II. THE CCM AND THE PCI-DSS

In [1], we have explained the main purpose of the norm PCI-DSS, its strength and its weaknesses linked to the cloud domain; the referential is rich but it is not adapted to the cloud environment.

The Cloud Security Alliance's CCM is a rich source of cloud security best practices designed as a framework to provide fundamental security principles to cloud vendors and cloud customers. It provides a controls framework that gives detailed understanding of security concepts and principles that are aligned to the Cloud Security Alliance guidance in 16 domains (latest version 3.0.1) [4]. This tool provides the holistic adherence to the vast and ever evolving landscape of global data privacy regulations and security standards.

The CCM serves as the basis for new industry standards and certifications. It is the first ever baseline control framework specifically designed for managing risk in the Cloud Supply Chain:

- Addressing the inter- and intra-organizational challenges of persistent information security by clearly delineating control ownership.
- Providing an anchor point and common language for balanced measurement of security and compliance postures.

 ISBN: 978-1-61208-427-5

The PCI-DSS is a broadly accepted set of policies and procedures intended to optimize the security of credit, debit and cash card transactions and protect cardholders against misuse of their personal information. Therefore, it is not possible for a client to leverage the benefits of cloud systems without jeopardizing security, and mainly Card Data.

In this paper, we focus on creating for each topic on CCM list a matching PCI-DSS requirement in order to get a series of checklists on what the client could depend on to verify the trustworthiness of the Cloud before deciding to outsource.

III. THE DOMAINS OF APPLICATION

In our work, we focused on 4 main areas (domains) because they represent a basis for any tenant to check and control Cloud before deciding to outsource or not. Figure 1 shows the four domains, which are Network and Transport security, Data Security, Application and interface security, and Business Continuity management:

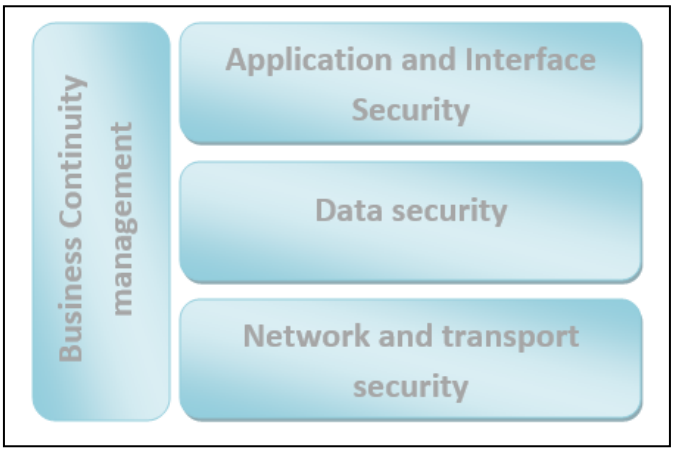


Figure 1. The domains developed in the checklist

- **Network and Transport security**: These controls allow verifying the security of the Card Data on network while it's transmitted. It is essential for the tenant to check this aspect scrupulously before deploying on the Cloud.
- **Data Security**: These controls allow verifying the security of the Card data and preventing it from any leakage.
- **Application and interface security**: These controls aim to ensure that any Application and Programming Interface (APIs) is designed, developed, deployed and tested respecting the PCI-DSS norms in order to avoid any leakage.
- **Business Continuity management**: These controls aim at insuring the business continuity of the activities in any issue or disaster. The client should be sure that the activity could continue without any deterioration.

In the next section, we describe the checklist developed with an exhaustive questionnaire as a tool for any Cloud specialist to verify the compliance of a cloud and its readiness to outsource or not.

IV. THE CHECKLIST MATRIX

Our work, as described above, is developing a checklist based on 4 domains and 33 controls. Each control addresses a part of securing Transaction payment on the Cloud. In the first part, we describe each control and in the second one, we present a small extract of the Cloud Checklist. For the full and exhaustive Checklist, as the document size is limited, we suggest to refer to the authors.

A. Network security

1) Network Security (Infrastructure & Virtualization Security)

In this control, the auditor must ensure that:

- The Network environments and virtual instances are designed and are configured to restrict and monitor traffic between trusted and untrusted connections.
- The configurations of the Network are reviewed at least annually, and are supported by a documented justification for use for all allowed services, protocols, and ports, and compensating controls.

2) Network Architecture (Infrastructure & Virtualization Security)

In this control, the auditor must ensure that:

- The network architecture diagrams have clearly identified high-risk environments and data flows that may have legal compliance impacts.
- The technical measures are implemented and apply defense-in-depth techniques for detection and timely response to network-based attacks associated with anomalous ingress or egress traffic patterns (e.g., MAC spoofing and ARP poisoning attacks) and/or distributed denial-of-service (DDoS) attacks.

3) VM Security - vMotion Data Protection (Infrastructure & Virtualization Security)

In this control, the auditor must ensure that:

- The secured and encrypted communication channels are used when migrating physical servers, applications, or data to virtualized servers
- There is a network segregation from production-level networks for such migrations.

4) Wireless Security (Infrastructure & Virtualization Security)

In this control, the auditor must ensure, in order to protect wireless network environments, that:

- There are policies and procedures that restrict the use of the this technology,
- The supporting business processes and technical measures are implemented.

5) Standardized Network Protocols (Interoperability & Portability)

In this control, the auditor must ensure that:

- The provider uses secure standardized network protocols for the import and export of data and to manage the service,

 ISBN: 978-1-61208-427-5

- The provider makes available a document to consumers (tenants) detailing the relevant interoperability and portability standards that are involved.

6) Audit Logging / Intrusion Detection (Infrastructure & Virtualization Security)

In this control, the auditor must ensure that:

- The provider is adhering to applicable legal, statutory or regulatory compliance obligations
- The provider is providing unique user access accountability to detect potentially suspicious network behaviors and/or file integrity anomalies, and to support forensic investigative capabilities in the event of a security breach.

7) Encryption (Encryption & Key Management)

In this control, the auditor must ensure, for the use of encryption protocols for protection of sensitive data in storage and data in transmission, that:

- The Policies and procedures are established,
- The supporting business processes and technical measures are implemented, as per applicable legal, statutory, and regulatory compliance obligations.

8) Antivirus / Malicious Software (Threat and Vulnerability Management)

In this control, the auditor must ensure, in order to prevent the execution of malware on organizationally-owned or managed user end-point devices and IT infrastructure network and systems components, that:

- The policies and procedures are established.
- The supporting business processes and technical measures are implemented.

9) Configuration Ports Access (Identity & Access Management)

In this control, the auditor must ensure that the user access to diagnostic and configuration ports is restricted to authorized individuals and applications.

10) Independent Audits (Audit Assurance & Compliance)

In this control, the auditor must ensure that independent reviews and assessments are performed at least annually to ensure that the organization addresses nonconformities of established policies, standards, procedures and compliance obligations.

11) User Access Policy (Identity & Access Management)

In this control, the auditor must ensure, in order for ensuring appropriate identity, entitlement, and access management for internal corporate and customer (tenant) users with access to data and organizationally-owned or managed (physical and virtual) application interfaces and infrastructure network and systems components, that:

- The user access policies and procedures are established,
- The supporting business processes and technical measures are implemented.

12) Segmentation (Infrastructure & Virtualization Security)

In this control, the auditor must ensure that the Multi-tenant organizationally-owned or managed (physical and virtual) applications, and infrastructure system and network components, are designed, developed, deployed and configured such that provider and customer (tenant) user access is appropriately segmented from other tenant users.

B. Data Security & Information Lifecycle Management

1) Data Inventory / Flows

In this control, the auditor must ensure that the policies and procedures are established to inventory, document, and maintain data flows for data that is resident (permanently or temporarily) within the service's applications and infrastructure network and systems (in particular, providers shall ensure that data that is subject to geographic residency requirements not be migrated beyond its defined bounds)

2) Classification

In this control, the auditor must ensure that data and objects containing data are assigned a classification by the data owner based on data type, value, sensitivity, and criticality to the organization.

3) eCommerce Transactions

In this control, the auditor must ensure that the data related to electronic commerce (e-commerce) that crosses public networks is appropriately classified, and protected from fraudulent activity, unauthorized disclosure, or modification in such a manner to prevent contract dispute and compromise of data.

4) Handling / Labeling / Security Policy

In this control, the auditor must ensure that:

- The policies and procedures are established for labeling, handling, and the security of data and objects that contain data.
- The mechanisms for label inheritance are implemented for objects that act as aggregate containers for data.

5) Nonproduction Data

In this control, the auditor must ensure that the production data aren't replicated or used in non-production environments.

6) Ownership / Stewardship

In this control, the auditor must ensure that all data is designated with stewardship, with assigned responsibilities defined, documented, and communicated.

7) Secure Disposal

In this control, the auditor must ensure that any use of customer data in non-production environments requires

 ISBN: 978-1-61208-427-5

explicit, documented approval from all customers whose data is affected, and must comply with all legal and regulatory requirements for scrubbing of sensitive data elements.

C. Application & Interface Security

1) Application Security

In this control, the auditor must ensure that the APIs are designed, developed, deployed and tested in accordance with leading industry standards (e.g., OWASP [10] for web applications) and are adhered to applicable legal, statutory, or regulatory compliance obligations.

2) Customer Access Requirements

In this control, the auditor must ensure that prior to granting customer's access to data, assets, and information systems, all identified security, contractual, and regulatory requirements for customer access are addressed and are remediated.

3) Data Integrity

In this control, the auditor must ensure that the data input and output integrity routines (i.e., reconciliation and edit checks) are implemented for application interfaces and databases to prevent manual or systematic processing errors, corruption of data, or misuse.

4) Data Security / Integrity

In this control, the auditor must ensure, in order to guarantee protection of confidentiality, integrity, and availability of data exchanged between one or more system interfaces, jurisdictions, or external business relationships to prevent improper disclosure, alteration, or destruction, that:

- The policies and procedures are established,
- The supporting business processes and technical measures are implemented.

D. Business Continuity Management & Operational Resilience

1) Business Continuity Planning

In this control, the auditor must ensure if all business continuity plans are consistent in addressing priorities for testing, maintenance, and information security requirements, that a consistent unified framework for business continuity planning and plan development is established, documented and adopted.

2) Business Continuity Testing

In this control, the auditor must ensure that:

- The business continuity and security incident response plans are subject to testing at planned intervals or upon significant organizational or environmental changes.
- The incident response plans involve impacted customers (tenant) and other business relationships that represent critical intra-supply chain business process dependencies.

3) Datacenter Utilities / Environmental Conditions (Power / Telecommunications)

In this control, the auditor must ensure that datacenter utilities services and environmental conditions (e.g., water, power, temperature and humidity controls, telecommunications, and internet connectivity) are secured, monitored, maintained, and tested for continual effectiveness at planned intervals.

4) Documentation

In this control, the auditor must ensure that information system documentation (e.g., administrator and user guides, and architecture diagrams) is made available to authorized personnel, in order to:

- Configure, install, and operate the information system,
- Effectively use the system's security features.

5) Environmental Risks

In this control, the auditor must ensure that the physical protection, against damage from natural causes and disasters, is anticipated, designed, and have countermeasures applied.

6) Equipment Location

In this control, the auditor must ensure, in order to reduce the risks from environmental threats, hazards, and opportunities for unauthorized access, that the equipment are kept away from locations subject to high probability environmental risks and are supplemented by redundant equipment located at a reasonable distance.

7) Equipment Maintenance

In this control, the auditor must ensure, for equipment maintenance ensuring continuity and availability of operations and support personnel, that:

- The policies and procedures are established,
- The supporting business processes and technical measures are implemented.

8) Policy

In this control, the auditor must ensure, for appropriate IT governance and service management to ensure appropriate planning, delivery and support of the organization's IT capabilities supporting business functions, workforce, and/or customers based on industry acceptable standards (i.e., ITIL v4 and COBIT 5), that:

- The policies and procedures are established,
- The supporting business processes and technical measures are implemented.

9) Retention Policy

In this control, the auditor must ensure, for defining and adhering to the retention period of any critical asset as per established policies and procedures, as well as applicable legal, statutory, or regulatory compliance obligations, that:

- The policies and procedures are established,
- The supporting business processes and technical measures are implemented.

 ISBN: 978-1-61208-427-5

In Table I, we illustrate an extract of the developed checklist. For each domain (from the main four described above), and for each sub-domain, we developed the questions that the auditors should verify and also how to verify the condition.

TABLE I. EXAMPLE OF CONTROL MATRIX (EXTRACT)

PCI-DSS Requirements Correspondent	Question	Expected Testing	In place	Not In Place	Reserves
A.1. Network Security					
PCI-DSS v3.0 1.1.2 Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks	Does a current network diagram exists and that it documents all connections to cardholder data, including any wireless networks?	• Examine diagram(s) • Observe network configurations	☐	☐	☐
	Is the network diagram kept updated?	• Interview responsible personnel	☐	☐	☐
PCI-DSS v3.0 1.1.3 Current diagram that shows all cardholder data flows across systems and networks	Does the diagram show all cardholder data flows across systems and networks?	• Examine data-flow diagram • Interview personnel	☐	☐	☐
	Is the diagram kept current and updated as needed upon changes to the environment?	• Examine data-flow diagram • Interview personnel	☐	☐	☐
….	…	• …	☐	☐	☐
A.7. Encryption					
PCI-DSS v3.0 2.1.1 For wireless environments connected to the cardholder data environment or transmitting cardholder data, change ALL wireless vendor defaults at installation, including but not limited to default wireless encryption keys, passwords, and SNMP community strings.	Were Encryption keys changed from default at installation?	• Interview responsible personnel • examine supporting documentation	☐	☐	☐
	Are encryption keys changed anytime anyone with knowledge of the keys leaves the company or changes positions?	• Interview responsible personnel • examine supporting documentation	☐	☐	☐
….	…	• …	☐	☐	☐
A.9. Identity & Access Management: Configuration Ports Access					
PCI-DSS v3.0 1.2.2 Secure and synchronize router configuration files.	Are router configuration files secured from unauthorized access?	• Examine router configuration files	☐	☐	☐
	Are router configurations synchronized?	• Examine router configurations	☐	☐	☐
….	…	• …	☐	☐	☐
B.3. eCommerce Transactions					
PCI-DSS v3.0 4.2 Never send unprotected PANs by end-user messaging technologies (for example, e-mail, instant messaging, chat, etc.)	Are end-user messaging technologies used to send cardholder data? (verify that PAN is rendered unreadable or secured with strong cryptography whenever it is sent via end-user messaging technologies)	• Observe processes for sending PAN • Examine a sample of outbound transmissions as they occur	☐	☐	☐
	Is there a policy stating that unprotected PANs are not to be sent via end-user messaging technologies?	• Review written policies	☐	☐	☐
….	…	• …	☐	☐	☐
C.1. Application Security					
PCI-DSS v3.0 6.5 : Address common coding vulnerabilities in software-development processes as follows: • Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory. • Develop	Are developers required training in secure coding techniques based on industry best practices and guidance?	• Review policies and procedures for training • Interview personnel	☐	☐	☐
	Are developers knowledgeable in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory?	• Interview personnel • Examine records of training	☐	☐	☐

PCI-DSS Requirements Correspondent	Question	Expected Testing	In place	Not In Place	Reserves
applications based on secure coding guidelines.	Are processes to protect applications from the following vulnerabilities, in place?		☐	☐	☐

V. CRITICAL VIEW TO THE STANDARD PCI-DSS ON THE CLOUD

Many controls specifications in the 4 domains treated above are not specified in any requirement in the recent version 3.0 of the PCI-DSS norm. These control specifications are:

- Network and Infrastructure Services: this control specification aims verifying that the Business-critical or customer (tenant) impacting (physical and virtual) application and system-system interface (API) designs and configurations, and infrastructure network and systems components, is designed, developed, and deployed in accordance with mutually agreed-upon service and capacity-level expectations, as well as IT governance and service management policies and procedures.
- Equipment Power Failure: this control specification aims verifying Information security measures and redundancies are implemented to protect equipment from utility service outages (e.g. power failures and network disruptions).
- Impact Analysis: this control specification aims verifying that there is a defined and documented method for determining the impact of any disruption to the organization that must incorporate the following:
    - Identify critical products and services
    - Identify all dependencies, including processes, applications, business partners, and third party service providers
    - Understand threats to critical products and services
    - Determine impacts resulting from planned or unplanned disruptions and how these vary over time
    - Establish the maximum tolerable period for disruption
    - Establish priorities for recovery
    - Establish recovery time objectives for resumption of critical products and services within their maximum tolerable period of disruption
    - Estimate the resources required for resumption.

In the next steps, we will evaluate this new framework by applying it on a real Card platform outsourced on the Cloud and we check its vulnerability and resilience. Afterward, we continue our work by focusing on developing recommended requirement for PCI-DSS for these control specification that could be added in the next update version of the norm.

VI. CONCLUSION AND FUTURE WORK

The goal of the PCI-DSS is to protect cardholder data that is processed, stored or transmitted by providers, issuers or merchants. The security controls and processes required by PCI-DSS are vital for protecting cardholder account data. With all the advantages that give Cloud, Issuers, and Merchants and any other service providers involved with payment card processing must insure that the platform virtually and physically is sufficiently protected.

In this paper, we have developed an exhaustive checklist as a tool for any card stakeholder who wants to outsource a part or the whole card processing in a Cloud. In the next steps of our work, we will focus on evaluating the robustness of this framework by applying it on a real application of Card Transaction Platform on the Cloud environment. Afterward, we will develop recommended requirements for PCI-DSS necessary for the Cloud Environment and we will release a new Self-Assessment Questionnaire as a reference for a Qualified Security Assessor to check in the Cloud environment.

REFERENCES

[1] H. El Alloussi, L. Fetjah, and A. Chaichaa, "Securing the Payment Card Data on Cloud environment: Issues & perspectives", International Journal Of Computer Science and Network Security, Vol. 14, no. 11, Nov. 2014, pp. 14-20, http://paper.ijcsns.org/07_book/html/201411/201411003.html.

[2] PCI Security Standards Council, "Requirements and Security Assessment Procedures", Version 3.0, November 2013, https://www.pcisecuritystandards.org [retrieved: May, 2015].

[3] PCI Security Standards Council, Summary of Changes from PCI-DSS Version 2.0 to 3.0", November 2013, https://www.pcisecuritystandards.org [retrieved: May, 2015].

[4] Cloud Special Interest Group (PCI Security Standards Council), "PCI-DSS Cloud Computing Guidelines", February 2013, https://www.pcisecuritystandards.org [retrieved: May, 2015].

[5] PCI Security Standards Council, "Payment Card Industry (PCI), Data Security Standard (DSS) and Payment Application Data Security Standard (PA-DSS), Glossary of Terms, Abbreviations, and Acronyms", Version 3.0, January 2014, https://www.pcisecuritystandards.org.

[6] Cloud Security Alliance (CSA), "CCM 3.0.1", https://cloudsecurityalliance.org/research/ccm/ [retrieved: May, 2015].

[7] G. Ataya, "PCI-DSS audit and compliance". In information security technical report 15 (2010) 138 -144.

[8] H. Rasheed, "Data and infrastructure security auditing in cloud computing", In International Journal of Information Management 34 (2014) 364–368.

[9] W. Spangenberg, "PCI Compliance in the Cloud: What are the Risks?", http://www.ioactive.com/pdfs/PCIComplianceInTheCloud.pdf.

[10] The Open Web Application Security Project (OWASP) Vulnerable Web Applications Directory Project, https://www.owasp.org/index.php/OWASP_Vulnerable_Web_Applications_Directory_Project, March 2015.

[11] G. Parann-Nissany, "Introduction to PCI-DSS and the Cloud", Sep 2013, http://www.infoq.com/articles/cloud-pci-compliance.

 ISBN: 978-1-61208-427-5

[12] J. P. de Albuquerque and P. L. de Geus. “A Framework for Network Security System Design”, WSEAS Transactions on Systems, Piraeus,Greece, v. 2, n. Issue 1, 2003, p. 139-144.

[13] N. Carr, “The Big Switch: h: Rewiring the World, from Edison to Google”, W.W. Norton & Co., NY, 2008.

[14] A. Toffler, “The Third Wave”, Bantam (1980).

[15] The ISO 27000 Directory, http://www.27000.org/, [retrieved: May, 2015].

[16] ISACA Global Organization/ COBIT, http://isaca.org/cobit.

[17] The National Institute of Standards and Technology, http://www.nist.gov/, [retrieved: May, 2015].

[18] The Technology Policy Division of the Financial Services Roundtable, http://www.bits.org, [retrieved: May, 2015].

[19] Generally Accepted Privacy Principles, https://www.cippguide.org/2010/07/01/generally-accepted-privacy-principles-gapp/, [retrieved: May, 2015].

[20] Health Insurance Portability and Accountability Act (HIPAA), http://www.ohii.ca.gov/calohi/PrivacySecurity/HIPAA.aspx, [retrieved: May, 2015].

[21] Jericho Forum, http://www.jerichoforum.org, [retrieved: May, 2015].

[22] North American Electric Reliability Corporation- Critical infrastructure protection, http://www.nerc.com/, [retrieved: May, 2015].

 ISBN: 978-1-61208-427-5

Ceremony Analysis Meets Verifiable Voting: Individual Verifiability in Helios

Taciane Martimiano*, Eduardo dos Santos, †, Maina Olembo‡, Jean Everson Martina*,
Ricardo Alexandre Reinaldo de Moraes,*
*Universidade Federal de Santa Catarina
Florianopolis - SC - Brazil
Email: taciane.m@inf.ufsc.br, jean.martina@ufsc.br, ricardo.moraes@ufsc.br
†University of Oxford - Oxford - United Kingdom
Email: eduardo.dossantos@cybersecurity.ox.ac.uk
‡Technische Universitt Darmstadt - Darmstadt - Germany
Email: maina.olembo@cased.de

***Abstract*—The Helios verifiable voting system offers voters an opportunity to verify the integrity of their individual vote, that it is cast, and included in the final count, as intended. While not all voters have to verify, these steps can be cumbersome for those who aim to carry them out. Therefore, new verification processes have been proposed in order to improve usability. Voters can use a web-based verifier provided by any one of several independent verification institutes, or a smartphone app developed and provided by these institutes. In this work, we describe these verification processes as ceremonies, and thus model the human peer's interaction. We undertake a security analysis applying an adaptive threat model suited to analysing human-device and human-human channels. More realistic threats on these channels are identified, compared to those from an analysis using a Dolev-Yao attacker.**

Keywords–Voting; Threat models; Security Ceremonies.

I. INTRODUCTION

In order to engender voter trust in electronic voting, cryptographic voting systems that offer verifiability while maintaining vote secrecy have been proposed, and continue to gain ground. The Scantegrity II end to end verifiable voting system was used in a governmental election [1] and a modified version of the Prêt à Voter system was used in the 2014 Victoria State elections in Australia [2]. In this space, Helios [3][4], an open-source, verifiable, Internet-based voting system, stands out for its continued use, primarily in academic contexts, for example, in 2009, to elect the university president at the Université Catholique de Louvain [4]. It was also used in the 2013 Princeton University undergraduate student government elections [5], and to elect the Board of Directors of the International Association for Cryptologic Research (IACR) [6].

With the use of Helios, it is assumed that voters can and will verify their votes to ensure vote integrity [3]. While it is not known whether this assumption is true for the elections where Helios has been used, findings from expert reviews [7] and user studies [8] suggest that this is not likely to be the case, due to the cumbersome nature of the verification process. Usability improvements to the Helios voting interfaces, with a specific focus on the verification aspect, have been proposed to ensure that this assumption can be met. These improvements involve the voter using verifiers provided by trusted institutes. These verifiers are available in two forms: either accessible to the voter through the institutes' web page, or via download and installation on the voter's smartphone as an app. In this work, we analyse the security implications of these improvements, a practice recommended for usable security [9]. The focus is on verifiability and integrity. We apply ceremony analysis [10] using the adaptive threat model provided by Carlos *et al.* [11], which is appropriate to analyse the human-device and human-human channels. Following standard practice, a Dolev-Yao adversary [12] is assumed on the device-device channels.

A. Contribution

Our findings show that:

1) No threats to secrecy are present when the voter uses the smartphone app to verify;
2) Reputation attacks might be carried out to undermine the institutes participating in elections. In such cases, voter education on necessary steps is required;
3) Semi-formal verification can be applied to election ceremonies.

We discuss the implications of these findings for voting and verification in Helios.

B. Related work

Several extensions to the Helios voting protocol have been proposed, focusing on providing everlasting privacy [13], privacy and correctness [14], and preventing attacks against privacy [15]. Zeus [16] is a verifiable voting and counting system developed based on the Helios version in [3]. The authors propose that the voter enters an *audit code* to indicate that a submitted vote should be verified. However, no analysis of the security implications of these modifications is provided. A variant of Helios that prevents ballot box stuffing is proposed in [14]. Comparatively, the research we report in this paper analyses the security of two proposals made to improve the usability of verification in Helios, in order to ensure that voters can indeed verify that their ballots are cast, and counted in the final tally, as intended.

Carlos *et al.* [11] proposed an adaptive threat model and applied it to analyse the Bluetooth pairing protocol. In our work, we apply their model to a new domain - verifiable voting.

This paper is structured as follows: Section II brings the necessary background on the Helios protocol, analysis on security ceremonies and the adaptive threat model used in this work. Section III shows the ceremony for the institutes' website proposal and its analysis. Section IV presents the ceremony for the app proposal and its analysis. Finally, Section V has our final remarks and conclusions.

 ISBN: 978-1-61208-427-5

II. BACKGROUND

In this section, we first describe the Helios voting protocol and then we provide background information on the design and analysis of ceremonies. Moreover, we introduce the adaptive threat model and the security criteria applied in this work.

A. The Helios voting protocol

We describe the Helios voting protocol, focusing only on those aspects that are relevant to verifiability and that are necessary to understand the proposals presented in later sections.

In order to vote using Helios, the voter downloads the Helios *ballot preparation system* (BPS)[17] onto his web browser. He indicates the candidate(s) of his choice on the ballot. The BPS encrypts these choices (i.e. the vote) and commits to the encryption by displaying a hash value, which we refer to as a check-code. The voter should record the check-code displayed if he plans to verify. At this point, the voter makes a choice to either submit this encrypted vote to the public web bulletin board or to challenge the voting system, verifying whether the vote has been correctly encrypted.

If the voter decides to verify, he interacts with the Helios *ballot verifier system* (BVS). The BPS displays the candidate(s) and the randomness used for encryption. The voter selects and copies this information to the voting device's clipboard and pastes it into the BVS, which BPS opens in a new web browser window. The BVS encrypts the corresponding candidate and generates the hash value of this encryption. This hash value is displayed together with the candidate(s) contained in the vote received earlier. In order to complete the verification process, the voter needs to confirm that the check-code displayed by the BVS matches the one displayed earlier by the BPS. Additionally, he needs to confirm that the vote is correct. If both these conditions are met, the voter is assured that the system correctly encrypted the vote in this instance. He can repeat the verification process several times until he is satisfied that the system is behaving correctly. Once votes are verified they can no longer be submitted to the public web bulletin board as the voter could easily prove how he voted using the revealed randomness. Thus, new randomness is required. As the BVS learns the content of the encryption, the use of test votes that differ from the final vote has been recommended [8], to avoid the BVS computing intermediate results.

If the voter chooses to submit his vote to the public web bulletin board, he is prompted to authenticate himself, and his encrypted vote is then posted on to the public web bulletin board together with the check-code. To verify that the encrypted vote is correctly stored on the voting server or public web bulletin board, the voter needs to confirm that the check-code appears on the public web bulletin board next to his name [3], or some pseudonym [4]. It is only necessary to do this once as the Helios threat model assumes that auditors continuously observe the bulletin board preventing malicious behaviour.

B. The design and analysis of ceremonies

Ceremonies extend protocols by including human peers and allowing the detection of otherwise undetectable security flaws [10]. In protocols, all the human actions are modelled as assumptions and, when the protocol is implemented, these assumptions can result in user interactions that are unrealistic or not well-specified. In ceremonies, additional channels are available to model the interaction of human peers to other peers in the system, namely, the human-human (HH) channel and human-device (HD) channel, besides the device-device (DD) channel from the protocol structure.

To take these additional channels into account, we analyse ceremonies using the adaptive threat model proposed by Carlos *et al.* [11]. This adaptive threat model uses the Dolev-Yao (DY) attacker's set of capabilities, by dynamically adding or removing capabilities from the whole set. Doing so helps in identifying cases where overly stringent requirements are placed on users. While such requirements are motivated by security concerns, they are likely to negatively impact usability. Understanding the correct threat model the user is subject to, when interacting in a ceremony, will prevent him from being overloaded with unrealistic scenarios and guarantee that important security properties will hold [11].

The analysis process begins with the establishment of channels present at the ceremony. This involves listing the human nodes and devices involved, identifying which of these nodes exchange messages and which type of communication channel they represent (i.e., HH, HD or DD). Thus, it is possible to analyse the impact of an attacker's capabilities in each channel. The attacker's goal is to learn the contents being exchanged among nodes. The DY threat model defines abilities that allow the attacker to achieve his goal. Therefore, we observe which approaches the attacker can use to stop or modify messages, create and send messages of their own knowledge, etc. in order to obtain a realistic threat model that includes the profiles of the attackers associated to each channel. For example, if the attacker has access to a given cryptographic key and intercepts messages encrypted with that key, he will be able to decrypt and learn the contents of these messages, compromising the safety of the messages shared by that channel. Interestingly, following the adaptive threat model of Carlos *et al.*, we have a realistic and specific threat model to each ceremony, given its participants, channels and environments to which it will be subject to.

C. An adaptive threat model

Dolev and Yao [12] formalised the attacker model introduced by Needham and Schroeder [18], giving the attacker absolute control of the network, such that the attacker can copy, replay, alter and create messages. However, he cannot perform cryptanalysis. Based on [11], the Dolev-Yao (DY) attacker has the following set of capabilities: Eavesdrop, Initiate, Atomic Break Down, Block, Crypto, Fabricate, Spoof, Reorder, Modify and Replay [11].

Assumptions: We present assumptions of the original Helios system, as well as our assumptions regarding the entities involved in the ceremony, and the ceremony itself.

The *attacker* lies only on the channel as is the usual Dolev-Yao assumption. Further, he cannot control more than one device-device channel.

Any participating *verification institute* is trustworthy as any malicious behaviour would lead to a loss of reputation. We are not considering denial of service attacks.

The *voter* is an honest peer in the ceremony as a dishonest voter can easily prevent a ceremony from concluding correctly.

 ISBN: 978-1-61208-427-5

We also do not consider coercion. Thus, the cases where the attacker is the voter are not included.

Finally, the *ceremony* is assumed to have only one entry and one exit point, so the voter is expected to follow all the steps provided in the ceremony he is executing.

We recognise some of the assumptions are weak. However, our main concern for this work is to establish the simplest version of the presented scenarios. Analysing more complex variations are left for future work.

1) Security criteria: As the adaptive threat model will be applied to the electronic voting context, we define necessary security criteria adapted from Neumann *et al.* [19].

A number of security properties are considered important in the electronic voting context. In this work, we concentrate on verifiability and integrity, likely the most important properties for elections conducted over a remote channel.

Integrity: The sum of all participating voters' submitted votes (votes submitted to and stored on the voting server or the public web bulletin board) matches the declared election result.

Integrity violations must not go undetected [20]. From this requirement, we obtain the definition of verifiability.

Verifiability: Property in which the voter assures himself of the integrity of the individual vote and the public is assured of the integrity of the election result. Verifiability consists of evidence of the following aspects being provided:

- The vote correctly represents the voter's choice;
- The vote has been stored on the voting server or public web bulletin board as it was cast by the voter;
- All valid votes on the public web bulletin board are tallied without modification.

III. Using a web-based ballot verifier

We summarise the processes that voters would carry out using a web-based ballot verifier provided by the trusted institutes. We analyse these processes using the adaptive threat model and briefly compare our results to those obtained in the case of a DY attacker, and close this section with a discussion of the results.

A. Proposal

The message flow for this proposal is seen in Figure 1. The text below the arrows identifies the channels under consideration. V refers to Voter, B to Booth, I to Institute, A to App and BB to Bulletin Board.

The voting process is similar to that described in subsection II-A. Note that differences in the voting and verification processes are reported in [7]. A relevant difference for this ceremony is that the voter enters the URL into the address bar in order to open the election website. He receives the voting credentials via postal mail, rather than clicking on a URL in the invocation email as in the original Helios.

We therefore concentrate on the verification processes where differences emerge between the original Helios and across the proposals presented in this work. In order to verify that his vote is correctly encrypted on the voting device, a voter first needs to record the check-code displayed by the BPS (see message 9 in Figure 1).

The voter then expresses to BPS his intention to verify, in message 10, views the verification institutes that are available in message 11, then selects an institute that he trusts, in message 12. He is re-directed to the selected institute's verification web page, in a new browser window. The BPS transmits the information necessary for verification, that is Vote + R and accompanying proofs to the selected institute, in message 13. Since the vote is transmitted to the institute, the case considered here is that the voter verifies a 'test vote', that is, one that is not equivalent to the final vote that he will cast.

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$
10.	V	$\xrightarrow[HD_1]{}$	B	:	"*Check encrypted vote*"
11.	B	$\xrightarrow[HD_1]{}$	V	:	Verifying Options Page
12.	V	$\xrightarrow[HD_1]{}$	B	:	I
13.	B	$\xrightarrow[DD_1]{}$	I	:	Vote + R
14.	I	$\xrightarrow[HD_2]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$ + Vote

Figure 1. Verifying vote using institutes' website

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$
10.	V	$\xrightarrow[HD_1]{}$	B	:	"*Log in to cast vote*"
11.	B	$\xrightarrow[HD_1]{}$	V	:	Login Page
12.	V	$\xrightarrow[HD_1]{}$	B	:	VoterID + Password
13.	B	$\xrightarrow[DD_3]{}$	BB	:	$\{Vote\}_{Epk}$
14.	B	$\xrightarrow[HD_1]{}$	V	:	Vote is submitted
15.	V	$\xrightarrow[HD_1]{}$	B	:	I
16.	B	$\xrightarrow[DD_1]{}$	I	:	$H\{\{Vote + R\}_{Epk}\}$
17.	I	$\xrightarrow[HD_2]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$ + VoterID

Figure 2. Submitting final vote using institutes' website

The institute will compute the check-code using the information it receives from the booth, and displays this, along with the vote it received, to the voter (message 14). The voter now needs to confirm that the two check-codes match (H_{BPS} = H_{BVS_I}), and that the vote displayed by the institute matches his selection on the ballot.

 ISBN: 978-1-61208-427-5

Next, we describe the process for the voter to verify that his vote is correctly stored on the voting server or public web bulletin board. This message flow is shown in Figure 2. The voter records the check-code displayed to him, in message. He then logs in to submit his vote, in messages 10 - 12. Upon successful authentication, the booth submits the vote to the bulletin board, in message 13. The voter would select an institute from several options displayed after he submits his vote, in message 15. A new web page would open. He would enter the recorded check-code information and view the result returned by the institute, in message 17.

Note that the institute additionally needs to display the Voter ID in its response to the voter, in order to prevent a successful clash attack [21], where different voters are shown the same check-code when they verify that their vote has been stored on the bulletin board. This can only be detected if the Voter ID is returned as it is a unique value.

B. Analysis

We apply the adaptive threat model in our ceremonies so we can conclude which scenarios are realistic and whether the attacker succeeds or not in his attempts to corrupt the system. For more specific and detailed information about the notation, formulae and semantics of the proofs, see Carlos et al[11].

1) Preliminaries: With the full Dolev-Yao (DY) attacker capabilities in mind and applying the framework proposed by Carlos *et al.* [11], we can evaluate our ceremonies against a less powerful and more realistic variation of such an attacker. Thus, we analyse the threat model each of the communication channels is subject to in each of the scenarios studied. We describe the adaptive threat model for each of the ceremonies and compare the results to the DY threat model. In the latter case, all the communication channels are under a full DY attacker.

In the adaptive threat model, only the device-device (DD) channel is under a full DY attacker, while the human-device (HD) channel is under a DY-E attacker. DY-E means that the attacker has all the DY capabilities, but lacks the eavesdrop capability. This capability is excluded since we are considering controlled environments, where the voter does not need to check around if there is someone eavesdropping his actions.

Considering the HD channel, we assume there is a human being (and not a machine pretending to be a human peer) communicating with a device. Thus, the voter interacts with his device (for example, looking at his computer screen or typing in the keyboard). A DY - E attacker is not able to compromise the secrecy of the messages sent through HD channels. This assumption is justified because the voter has control of his computer, thus limiting the attacker's actions. Therefore, once the attacker is not able to learn any voter's information, he can only apply his other capabilities over the knowledge he already has. For example, even if the attacker fabricates messages or uses his crypto capability, he can only use his own knowledge, which poses no threat to the voting and verifying ceremonies. We show proofs informing the knowledge each peer of the system and the attacker have through the set knows(Y), representing the set of knowledge of an agent Y in the ceremony [11].

We now move to the actual analysis of the ceremonies involving the institutes.

2) Test vote is correctly encrypted on the voting device: Based on Figure 1:

If the messages M1 to M12, and message M14 are run against a DY-E attacker, and message M13 is run against a DY attacker, the attacker (Att) can prevent the institute I from learning Vote + R and instead send $Vote_{att}$ + R instead, where $Vote_{att}$ is chosen by the attacker.

$$\frac{(M_{1...12,14} \cup DY - E) \wedge (M_{13} \cup DY)}{\begin{array}{r} Vote \wedge R \wedge Vote_{att} \in knows(Att) \wedge \\ Vote \in knows(B) \wedge \\ (Vote + R) \notin knows(I) \wedge (Vote_{att} + R) \in knows(I) \end{array}}$$

a) Proof: Assume the attacker *Att* initiated two simultaneous pairing sessions between the booth B and the institute I in message 13. *Att* uses his block, atomic breakdown, fabricate and initiate capabilities in this message, preventing I from learning the correct vote and randomness information, that is (Vote+R), forcing it to receive ($Vote_{att}$ +R) instead.

3) Final vote is correctly stored on the voting server or public web bulletin board: Based on Figure 2:

If the messages M1 to M12, M14, M15 and M17 are run against a DY-E attacker, and message M13 and M16 are run against a DY attacker, the attacker (Att) can prevent the bulletin board BB from receiving the correct $\{Vote\}_{Epk}$. Att can also prevent the institute I from learning $H\{\{Vote + R\}_{Epk}\}$. Instead, he sends altered information $\{Vote_{att}\}_{Epk}$ and $H_{att}\{\{Vote_{att} + R_{att}\}_{Epk}\}$ to the bulletin board BB and the institute I, respectively, where $Vote_{att}$ is chosen by the attacker. Then, the attacker uses his crypto capability to generate the $\{Vote_{att}\}_{Epk}$ information and his fabricate capability to generate $H_{att}\{\{Vote_{att}\} + R_{att}\}_{Epk}\}$.

$$\frac{(M_{1...12,14,15,17} \cup DY - E) \wedge (M_{13,16} \cup DY)}{\begin{array}{r} \{Vote\}_{Epk} \wedge \{Vote_{att}\}_{Epk} \wedge H\{\{Vote + R\}_{Epk}\} \wedge \\ H_{att}\{\{Vote_{att} + R_{att}\}_{Epk}\} \in knows(Att) \wedge \\ Vote \in knows(B) \wedge \{Vote\}_{Epk} \notin knows(BB) \wedge \\ \{Vote_{att}\}_{Epk} \in knows(BB) \wedge \\ H\{\{Vote + R\}_{Epk}\} \notin knows(I) \wedge \\ H_{att}\{\{Vote_{att} + R_{att}\}_{Epk}\} \in knows(I) \end{array}}$$

a) Proof: We assume the attacker *Att* initiated two simultaneous pairing sessions between booth B and the public web bulletin board BB maintained by institute I. The attacker Att uses his block, fabricate and initiate capabilities (message 13 in Figure 2) and sends to the bulletin board BB $\{Vote_{att}\}_{Epk}$, instead.

Note that the attacker can know the existing votes and the public key of the election, however the attacker cannot know the randomness information R.

C. Results

If we consider scenarios with the DY threat model, the attacker has total control of all channels and is able to manipulate the voter in the whole voting process. In these scenarios, the attacker intercepts all messages, sending messages in his knowledge to booth B, institute I, and bulletin board BB,

 ISBN: 978-1-61208-427-5

instead of the original messages. At the same time, he displays to the voter the right content, pretending to be the legitimate entities. Therefore, the voter is led to believe that his vote was encrypted, submitted and stored as intended when this is not the case. Nevertheless, such a scenario is highly unlikely to happen in real world situations, as the HD channel limits the attacker's actions. Furthermore, by involving the human peer, it is difficult for the attacker to control this channel and the information being exchanged without being noticed.

A scenario that is realistic and feasible, involves the attacker intercepting messages on the DD channel. Therefore, the institute receives altered information and calculates a different check-code from the one expected by the voter. The voter then no longer trusts the institute, believing it to be unreliable. This result highlights the need for multiple institutes to be available, providing verification services to voters. The voter is free to verify with several other institutes. If these subsequent checks also return a failed result, he can then contact the election commission.

Analysing the two ceremonies presented above, we can see Vote+R being sent without any encryption in the first ceremony while the second one contains the vote encrypted with the public key of the election (E_{pk}). From this we can conclude that secrecy does not hold in the ceremony for the test vote represented in Figure 1. Secrecy does hold when the voter decides to cast his final vote, represented in Figure 2.

We can conclude this given the fact that even when the DY attacker intercepts message 13 in the DD channel, he only sees a check-code that does not give any information about the vote or the randomness information used. The attacker thus cannot know the vote.

IV. Using a smartphone app and QR codes

We describe the processes that voters would carry out to verify with a verifier installed as an app on the smartphone. With this proposal, the voter has a way to verify that is separate from the voting device. Thus, there is no longer a need to trust the voting device with respect to integrity. Additionally, it uses a device that is in the voter's possession and that he likely trusts (with respect to secrecy and integrity). We analyse these ceremonies using an adaptive threat model, and compare our findings to those using a DY attacker. We conclude this section with the results of our analysis.

A. Proposal

The message flow for this proposal is shown in Figure 3. The BPS displays a QR code containing the check-code in addition to the human-readable value, in message 9. The voter opens the smartphone app and scans the QR code containing the check-code, in messages 10 and 11. This check-code will be stored by the app for later use during the verification process. The voter then expresses his intention to verify to the voting booth, in message 12.

He scans a second QR code, in message 15, and the app computes the check-code comparing it to the first check-code. It then informs the voter that the check-codes match (in a success scenario), and prompts him to confirm that the displayed vote matches his initial input on the ballot, in messages 16 - 17. This is an implementation of a *forcing function* [22] preventing the voter from proceeding without confirming that his vote is correct.

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	Check-code + $QRCode^*$
10.	V	$\xrightarrow[HD_3]{}$	A	:	"*Open/Scan*"
11.	B	$\xrightarrow[DD_2]{}$	A	:	QRCode
12.	V	$\xrightarrow[HD_1]{}$	B	:	"*Check encrypted vote with app*"
13.	B	$\xrightarrow[HD_1]{}$	V	:	QRCode Page
14.	V	$\xrightarrow[HD_3]{}$	A	:	"*Check that vote is correctly encrypted*"
15.	B	$\xrightarrow[DD_2]{}$	A	:	$QRCode^{**}$
16.	A	$\xrightarrow[HD_3]{}$	V	:	"*Check-codes match. Is this your vote?*" + Vote
17.	V	$\xrightarrow[HD_3]{}$	A	:	Confirm (Yes)
18.	V	$\xrightarrow[HD_3]{}$	A	:	"*Close*"

*Check-code is the hash $H\{\{Vote+R\}_{Epk}\}$. In messages 9 and 11, the QRCode has the check-code contents.

**In messages 13 and 15, the QRCode has $(Vote+R+E_{pk})$ information.

Figure 3. Verifying vote using institutes' app

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	Check-code + $QRCode^*$
10.	V	$\xrightarrow[HD_3]{}$	A	:	"*Open/Scan*"
11.	B	$\xrightarrow[DD_2]{}$	A	:	QRCode
12.	V	$\xrightarrow[HD_1]{}$	B	:	"*Log in to cast vote*"
13.	B	$\xrightarrow[HD_1]{}$	V	:	Login Page
14.	V	$\xrightarrow[HD_1]{}$	B	:	VoterID + Password
15.	B	$\xrightarrow[DD_3]{}$	BB	:	$\{Vote\}_{Epk}$
16.	B	$\xrightarrow[HD_1]{}$	V	:	Vote is submitted
17.	V	$\xrightarrow[HD_3]{}$	A	:	"*Check that vote is on bulletin board*"
18.	A	$\xrightarrow[DD_4]{}$	BB	:	$H\{\{Vote+R\}_{Epk}\}$
19.	BB	$\xrightarrow[DD_4]{}$	A	:	Confirmation + VoterID

*Check-code is the hash $H\{\{Vote+R\}_{Epk}\}$. In messages 9 and 11, the QRCode has the check-code contents.

Figure 4. Submitting final vote using institutes' app

In order to verify that a vote is correctly stored on the voting server or public web bulletin board (see Figure 4), the voter scans the first QR code containing the check-code as in the previous ceremony, in message 10. He logs in (messages 12 - 14) and the booth submits his vote upon successful authentication, in message 15. The voter instructs the app to check the public web bulletin board for the stored check-code, in message 17. The app does this by querying the public web bulletin board for the check-code, in message 18. In a success scenario, it displays a message to the voter that the check-code

was stored on the public web bulletin board. To prevent clash attacks [21], the app should return the accompanying voter ID as well. In a failure scenario, the voter will be informed that the check-code was not found. He can use other apps for verification. In order for the ceremony to terminate, the voter will be directed to contact the election commission should multiple checks with various apps return failed results.

B. Analysis

Before we present the analysis and results, we first review necessary considerations for the adaptive threat model.

1) Preliminaries: Although DD channels are usually under the full DY attacker, it is not realistic in the DD_2 channel (for example, message 11 in Figure 3). This channel characterises a 'visual channel' once there is no Bluetooth or connection of any kind between these devices. We are considering the ideal case where the smartphone has no virus or worms, for simplicity. Nonetheless, instructions regarding safety or the lack thereof could be given to the user, so he would be aware of the threat model he is subject to and decide whether continue or quit the remaining proceedings.

Channel DD_2 represents the scenario where the voter is using his smartphone to scan the QR code displayed by the computer. Both devices are considered to be in voter's possession, and not under the attacker's control. We have a similar situation as described for the HD channel (in Section III-B1). For example, it is clear that the attacker cannot block any contents passing through the DD_2 channel as this would imply the attacker blocking the computer screen from the voter and his smartphone. The same holds if the attacker tries to perform any of his other capabilities. Thus, for a successful attack, the attacker has to possess the voter's devices. This is only feasible if the voter leaves the devices unattended in the middle of the vote casting process.

Therefore, we consider the DD_2 channel as being DY - E, similar to the HD channel. Any weakened variation of the DY - E attacker (any combination of the capabilities of the full DY attacker, less eavesdrop) can be used because this attacker will not be effective. This is due to the fact that eavesdrop is the only capability which can compromise the secrecy of the voter's vote.

We now move to analyse each ceremony involving the app individually, using the adaptive threat model, and the DY threat model, respectively.

2) Verify vote is correctly encrypted on the voting device:: Based on Figure 3:

If all messages M1 to M18 are run against a DY-E attacker, the attacker cannot perform any significant attack with respect to secrecy and integrity.

$$\frac{M_{1...18} \cup DY - E}{\emptyset}$$

a) Proof: For this ceremony, we consider that the DD_2 channel (in messages 11 and 15 of Figure 3) is not under a full DY attacker. This means that these channels are a weakened variation of the DY threat mode,l because the scenario involves a computer displaying a QR code and communicating with the voter's mobile device. Thus, once the attacker is unable to eavesdrop on the communication, all the other capabilities the attacker has will not have an effect on the secrecy of the voter's vote in the ceremony. Therefore, as the attacker cannot possess the voter's devices and he cannot know the voter's vote, he can no longer perform any significant attack.

3) Final vote is correctly stored on the voting server or public web bulletin board: Based on Figure 4:

Consider messages M1 to M14, M16 and M17 are run against a DY-E attacker, and messages M15, M18 and M19 are run against a DY attacker. The attacker (Att) *can prevent the bulletin board BB from learning the correct values of* $\{Vote\}_{Epk}$ *and* $H\{\{Vote + R\}_{Epk}\}$.

$$\frac{(M_{1...14,16,17} \cup DY - E) \wedge (M_{15,18,19} \cup DY)}{\begin{array}{r} Vote \wedge Vote_{att} \wedge \{\text{Vote}\}_{Epk} \wedge \\ \{\text{Vote}_{att}\}_{Epk} \wedge \text{H}\{\{\text{Vote + R}\}_{Epk}\} \wedge \\ \text{H}_{att}\{\{Vote_{att} + R_{att}\}_{Epk}\} \in knows(Att) \wedge \\ Vote \in knows(B) \wedge \\ \{\text{Vote}\}_{Epk} \wedge \text{H}\ \{\{\text{Vote + R}\}_{Epk}\} \notin knows(BB) \wedge \\ \{\text{Vote}_{att}\}_{Epk} \wedge \text{H}_{att}\{\{Vote_{att} + R_{att}\}_{Epk}\} \in knows(BB) \end{array}}$$

a) Proof: We assume the attacker *Att* initiated two simultaneous pairing sessions between the booth B and the bulletin board BB (message 15 in Figure 4) and between the app A and BB (messages 18 and 19 in Figure 4). We continue to assume that the DD_2 channel has a DY-E attacker since it is a visual channel. The attacker *Att* uses his capabilities of block, fabricate and initiate in messages 15, 18 and 19 where *Att* sends to the bulletin board BB a different value of the encrypted vote and a different value of the check-code, instead of the original ones.

C. Results

If we consider the DY threat model, the attacker can manipulate the voter by manipulating the information displayed to him. Such a situation can be considered realistic (see Figure 4). However, it is highly unlikely to happen due to the fact that HD channels are secure under our assumption that the environment is controlled. In addition, we demonstrated it is unrealistic (Figure 3). This ceremony is more secure due to the presence of the visual channel, which limits the attacker's actions, once it has the same behaviour as on the HD channels. A very important contribution of the app proposal is that the test and final votes are both secret, when compared to the proposal that uses the institutes (where the test vote is sent in clear through a full DY channel). Such a contribution means that the security property of secrecy holds in the app ceremony and, as the messages are no longer interrupted and modified, we can conclude this ceremony also ensures integrity.

In the ceremonies involving verifying using the app, the attacker cannot control more than one DD channel [11]. Therefore, either the attacker controls the message between the booth B and the bulletin board BB (message 15 of Figure 4) or he controls the messages between the app A and the bulletin board BB (message 18 of Figure 4). When the attacker succeeds, the bulletin board BB does not display to the voter the expected confirmation (message 19 of Figure 4). In such a situation, the voter would be advised to contact the election commission.

 ISBN: 978-1-61208-427-5

V. CONCLUSION

Two verification proposals have been made to improve the usability of the individual verifiability processes in the Helios voting system. We have analysed the security of these proposals using a framework proposed by Carlos *et al.* [11], which uses an adaptive threat model. To this end, we considered the voting and verification processes in Helios as ceremonies in order to include the human peer's interaction in our analysis. The Dolev-Yao adversary model is shown to bestow unrealistic powers on the attacker. Hence, an adaptive threat model is applicable in analysing the voting and verifying ceremonies.

The proofs presented in this paper were subject to formal verification using the theorem prover SPASS and the same technique by Carlos et. al. Again due to space constraints they were not included but are available at:
https://github.com/tacianem/HeliosSpass

The first verification proposal involves the voter verifying using a web-based verifier provided by a trusted institute. Our results show the possibility of secrecy violations when the voter verifies that his vote is correctly encrypted on the voting device, and integrity violations when he also verifies that his vote is correctly submitted to the bulletin board. The secrecy violations would only arise if voters did not verify test votes. Integrity violations, on the other hand, would take the form of 'reputation' attacks, resulting in the voter mistrusting the institute as he detects that it displays incorrect information.

The situation is seen to improve with regard to the smartphone app. First, secrecy is maintained due to the presence of a visual channel, and because information is transmitted in encrypted form. Our results also show that no significant attack can occur when the voter verifies that his vote is correctly encrypted on the voting device. Integrity violations are as in the previous case, where reputation attacks would lead to a mistrust of the participating institute.

One can argue that the impact of the reputation attacks is low due to the availability of mitigating strategies. We highlight that integrity assurances in both verification proposals rest on the distribution of trust, that is, there are several options, whether web-based verifiers, or smartphone apps from trusted institutes, available for the voter to use. Should the verification process fail in one case, the voter can verify using other sources. Indeed, voters who do not want to trust any of the available institutes can use all the provided verification mechanisms. We acknowledge that this is not an ideal case with regard to usability, however, we note that it places less of a burden on the voter than would be the case if a more powerful adversary was considered.

The results of this work have highlighted several improvements that can be made to the Helios voting protocol. This will be the focus of future work. As the use of test votes is likely to have a negative impact on usability, we will make further improvements in this regard. One proposal is for the voter to enter some unique information known only to him. Verifying the presence of this information at a later stage assures him of the integrity of his submitted vote. Proposals will be made with the objective of balancing security and the expectations and abilities of the human peers in the ceremony.

VI. ACKNOWLEDGEMENT

Support by CAPES foundation Brazil.

REFERENCES

[1] D. Chaum and et al., "Scantegrity ii: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes," ser. EVT'08. USENIX Association, 2008.

[2] C. Burton, , and et al., "Using Prêt à Voter in Victorian State Elections," ser. EVT/WOTE'12. USENIX Association, 2012.

[3] B. Adida, "Helios: Web-based Open-Audit Voting," in Proceedings of the 17th Symposium on Security. Usenix Association, 2008, pp. 335 – 348.

[4] B. Adida, O. De Marneffe, O. Pereira, and J.-J. Quisquater, "Electing A University President using Open-Audit Voting: Analysis of Real-World Use of Helios," ser. EVT/WOTE'09. Usenix Association, 2009.

[5] Princeton, "Princeton Undergraduate Elections," accessed on 30th June, 2015. [Online]. Available: https://princeton.heliosvoting.org/

[6] IACR, "IACR Board of Directors 2013 Election and Referendum on Bylaws Amendments," accessed on 30th June, 2015. [Online]. Available: https://vote.heliosvoting.org/helios/elections/b36cbf0c-250a-11e3-89f4-46d2afa631be/view

[7] F. Karayumak, M. Kauer, M. M. Olembo, and M. Volkamer, "Usability Analysis of Helios - An Open Source Verifiable Remote Electronic Voting System," ser. EVT/WOTE'12. USENIX Association, 2011.

[8] F. Karayumak, M. Kauer, M. M. Olembo, T. Volk, and M. Volkamer, "User Study of the Improved Helios Voting System Interfaces," in STAST, 2011.

[9] M. A. Sasse and I. Flechais, "Usable Security: Why Do We Need It? How Do We Get It?" in Security and Usability: Designing Secure Systems That People Can Use. O'Reilly, 2005.

[10] C. Ellison, "Ceremony Design and Analysis," Cryptology ePrint Archive, Report 2007/399, 2007.

[11] M. C. Carlos, J. Martina, G. Price, and R. F. Custodio, "An Updated Threat Model for Security Ceremonies," in ACM Symposium on Applied Computing. ACM, 2013.

[12] D. Dolev and A. C. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. 29, 1983.

[13] D. Demirel, J. Van De Graaf, and R. Araújo, "Improving Helios with Everlasting Privacy Towards the Public," ser. EVT/WOTE'12. USENIX Association, 2012.

[14] V. Cortier, D. Galindo, S. Glondu, and M. Izabachene, "A Generic Construction for Voting Correctness at Minimum Cost - Application to Helios," Cryptology ePrint Archive, Report 2013/177, 2013.

[15] V. Cortier and B. Smyth, "Attacking and Fixing Helios: An Analysis of Ballot Secrecy," ser. CSF '11. IEEE Computer Society, 2011.

[16] G. Tsoukalas, K. Papadimitriou, P. Louridas, and P. Tsanakas, "From Helios to Zeus," EVT/WOTE'13.

[17] B. Adida, "Helios: Web-based open-audit voting."

[18] R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," ACM Press, 1978.

[19] S. Neumann, J. Budurushi, and M. Volkamer, Analysis of Security and Cryptographic Approaches to Provide Secret and Verifiable Electronic Voting. IGI Global, 2013.

[20] L. Langer, A. Schmidt, J. Buchmann, and M. Volkamer, "A Taxonomy Refining the Security Requirements for Electronic Voting: Analyzing Helios as a Proof of Concept," in ARES'10. IEEE, 2010.

[21] R. Küsters, T. Truderung, and A. Vogt, "Clash Attacks on the Verifiability of E-Voting Systems," in IEEE Symposium, 2012, pp. 395–409.

[22] D. A. Norman, The Design of Everyday Things. Basic Books, Inc., 2002.

 ISBN: 978-1-61208-427-5

Mobile Agent Security using Reference Monitor-based Security Framework

Sandhya Armoogum, Nawaz Mohamudally
Dept. Industrial Systems & Engineering
University of Technology, Mauritius (UTM)
La Tour Koenig, Mauritius
email: asandya@umail.utm.ac.mu

Nimal Nissanke
Emeritus Professor,
London South Bank University, London, UK
email: nissanke@gmail.com

***Abstract*— In distributed systems and in open systems such as the Internet, often mobile code has to run on unknown and potentially hostile hosts. Mobile code, such as a mobile agent is vulnerable when executing on remote hosts. The mobile agent may be subjected to various attacks such as tampering, inspection, and replay attack by a malicious host. Much research has been done to provide solutions for various security problems, such as authentication of mobile agent and hosts, integrity and confidentiality of the data carried by the mobile agent. Many of such proposed solutions in literature are not suitable for open systems whereby the mobile code arrives and executes on a host which is not known and trusted by the mobile agent owner. In this paper, we propose the adoption of the reference monitor by hosts in an open system for providing trust and security for mobile code execution. A secure protocol for the distribution of the reference monitor entity is described as well as a novel approach to assess the authenticity and integrity of the reference monitor running on the destination agent platform before any mobile agent migrates to that destination. This reference monitor entity on the remote host may provide several security services such as authentication, integrity and confidentiality of the agent's code and/or data.**

Keywords- Security; Mobile agents; Reference monitor, Trust

I. INTRODUCTION

During the last decade, there have been major changes in distributed computing. Programs are no longer constrained to execute on one machine. Code can now be migrated to other hosts for execution. The most well-known example of mobile code is the use of Java applets and JavaScript code in a Web browser. This form of code mobility is referred to as code on demand. Code mobility has many uses, i.e., vendors can use mobile code to reconfigure software, Microsoft uses mobile code to distribute software patches, mobile code can also be used to manage distributed system by performing load balancing [1]. Mobile agents represent a more sophisticated and powerful form of code mobility. A mobile agent is a program that can move from one host to another according to its own internal logic. Such mobile agents can have weak or strong mobility.

Mobile agent computing paradigm presents numerous advantages, compared to the traditional client-server based computing model, which include reduced network usage, better fault tolerance, adaptability to changes in the environment, and platform independence [2]. Thus, the mobile agent computing paradigm has become a natural and flexible way for implementing many applications on the network such as e-commerce and auctioning, network monitoring, real-time control systems and cloud computing. Recently, mobile agent computing proves to be very useful in the context of wireless sensor network (WSN) [3]. As such, the mobile agent computing paradigm provides a very intuitive and flexible approach to solving old and new problems arising in many areas of computing (e.g. intrusion detection [4], web crawling [5], Big Data analysis [6], searching and communications in Internet of Things [7] and e-learning [8] due to their mobility and autonomous nature, their ability to learn and adapt to changing environments, their ability to communicate and collaborate with one another to perform some complex task, and their ability to clone themselves, if needed.

However, for mobile agent applications to be widely adopted, some security issues have to be addressed, as mobile agent applications, mainly deployed in open environments, may possibly be exposed to attacks. Currently, the lack of an integrated security solution is a main drawback, which has to be tackled before the mobile agent computing paradigm is widely accepted by industry. Four threat categories have been identified in [9] as follows: (1) mobile agent attacking an agent platform (AP), (2) AP attacking a mobile agent, (3) a mobile agent attacking another agent on the AP, and (4) other entities attacking the agent system. Some practical solutions exist for securing AP against malicious mobile agents. However, securing the mobile agents against malicious AP is more challenging as the host has full control of the execution environment. Moreover, the mobile agent computing model violates some of the fundamental assumptions of conventional security techniques, i.e., programs runs on hosts which are trusted. Other important assumptions are the identity and intention assumptions [10]. Usually, when a program attempts some action, the program acts on behalf of a known user and it is assumed that the person intends the action to be taken. When mobile agents execute on unknown hosts, then neither identity or intention of the host is clear.

The fact that mobile agents may have to operate in such unknown and open environment requires the concept of trust of the hosting agent platforms despite that they may be previously unknown to the mobile agent's owner [11]. In this paper, we propose an approach for building trust in a mobile agent system by exploiting the reference monitor concept which provides trust of unknown APs and subsequently trust enhanced security. Such a reference monitor entity is obtained from a trusted third party (TTP) and can provide or arbitrate several security services resulting in a system which supports mobile agent applications in an open system.

 ISBN: 978-1-61208-427-5

The rest of this paper is structured as follows. Section II discusses related work. Section III presents the proposed security framework based on trust provided by the Reference Monitor for providing a trusted computing environment. Section IV discusses the integration and evaluation of the reference monitor entity in an agent platform. Section V presents the experimental setup used and results. Finally, Section VI presents the conclusions drawn and outlines future work.

II. RELATED WORK

Several security mechanisms exist to secure mobile agents as they execute on remote hosts. However, most of the existing schemes mainly allows the detection of integrity attacks on the mobile agent and none of these solutions provide a comprehensive integrated security framework for protecting mobile agents.

One approach for mobile agents to have comprehensive security against malicious platform involves allowing mobile agents to migrate to known and trusted APs only [12]. Then, agents are sent in encrypted form from one trusted AP to another, where they execute, often after authentication of the AP. However, such an approach seriously limits the agent's execution on a limited number of known and trusted APs. It is also not suitable for some applications like search agents (searching for information) on the internet, and mobile e-commerce agents where AP may not always be known and is against the notion of an open multi-agent system where new APs can be dynamically added or removed from an agent's itinerary.

Another proposed solution for providing comprehensive security to mobile code advocates the use of trusted, tamper-proof hardware which is not controlled by the local system and which supports secure mobile agents execution [13]. This secure trusted computing base on each AP, thus, provides the trusted environment for running critical code of the mobile agent. Here, mobile agents move from one trusted environment to another. Local resources on the system are accessed in a client-server mode. The system outside the trusted hardware has no access and thus cannot interfere with the execution of the mobile agent. The external system can only interact with the trusted tamper resistant hardware via some restricted interface. The main drawback of this approach is that every host has to be supplied with secure trusted hardware which is not a simple task as such hardware installation and maintenance may be expensive. Furthermore, the use of tamper-proof hardware may not scale-up efficiently and may be limited to highly security-sensitive mobile agent computing areas such as banks and stock markets. Moreover, the tamper-resistant devices could also become a performance bottleneck in the execution of mobile agent especially in cases where smart cards are used as a cheap alternative for providing hardware trusted computing base [14].

Another approach for protecting mobile agents involves code obfuscation, whereby the agent's program is made illegible and data hidden, thus rendering it difficult to read and modify the agent code, data and partial results [15]. An obfuscated mobile agent is like a black-box entity; it only permits AP to provide inputs and read outputs from the mobile agent. Thus, even if the mobile code runs on unknown and untrusted APs, it can maintain confidentiality and integrity. However, this technique often only provides limited code confidentiality as given time, the malicious AP will be able to de-obfuscate the code. In [16], the authors actually show that complete obfuscation is impossible. Furthermore, the malicious host could still re-execute (replay) the mobile agent several times so as to observe its reaction and guess its decision making strategy for example. Esparza, et al. [17] proposes to monitor the execution time on an AP. A longer than expected execution time on an AP is indication that the AP may be attempting to de-obfuscate the agent, modify the agent code, data and/or partial results or replay the mobile agent. The main drawbacks of the approach are that it requires the agent to return with the partial results from each host visited to the Home Agent Platform (HAP), i.e., HAP must be connected until the transaction is over. The application would then no longer support disconnected and asynchronous processing as promised by mobile agent technology, though it does provide security of mobile agent such as integrity, execution integrity, and detection of Denial of service (DoS) attack.

Finally, the use of mobile cryptography (also referred to as function hiding) which aims to offer provably strong protection to mobile agents against both modification and inspection attacks has been proposed. Mobile cryptography is such that a mobile agent program is encrypted into a ciphered executable program where it can execute on the untrusted AP while remaining in the ciphered form [18] [19]. The efficiency of this approach is unknown and to the best of our knowledge, there is no practical implementation. It is still unclear if such a scheme can be implemented for real-life applications. It is thus obvious that securing mobile agent remains a challenging research problem.

Trust is an important component of mobile code security [20]. When agents are executing in open environments on unknown and untrusted APs, if they could have some guarantees of trust, this would provide a basis for better security solutions. In [21], the authors describe a trust management architecture (MobileTrust) that can be developed to manage security related trust relationships explicitly and to make trust decisions. In such a system, trust management brings an improvement in security as by leveraging the trust knowledge gained on the past behaviours of other execution hosts, the mobile agents itinerary can be composed such as to minimize security attacks from potential malicious APs. Similar trust computation can allow the AP to evaluate the trustworthiness of a mobile agent from a specific agent owner. However, such a technique may not be suitable for all open environment applications. For example, in a mobile agent based e-commerce application, a mobile agent from an owner may not visit the same AP twice and thus the trust values of the current execution may never be useful. In [22], a trusted third party called a Clearing House is proposed to maintain trust information about APs. Before a mobile agent migrates to an AP, trust level associated with the AP may be found out. However, this solution heavily relies on the Clearing House which keep

 ISBN: 978-1-61208-427-5

tracks of the trustworthiness of each AP and also may not be suitable for applications where the mobile agent have a dynamic itinerary. Finally, such mechanisms of monitoring behaviour of AP may be unpractical since not all attacks can be detected and thus reported, especially breach in agent data and code confidentiality.

III. REFERENCE MONITOR BASED SECURITY FRAMEWORK

In this research work, we propose a practical and pragmatic technique for providing mobile agent security based on the reference monitor (RM) concept. Since its introduction, in the early 1970s, in the "Anderson Report" [23], the RM concept has been adopted for securing computer and network. This concept visualises a system component, called a *reference validation mechanism*, to be responsible for administering the system's security policy. It thus defines the requirements for implementing such a mechanism in a manner that ensures that malicious entities cannot circumvent policy enforcement [24]. The RM concept thus provides a trusted and verifiable security policy enforcement mechanism[25]. This is in line with the U.S Government's criteria for building secure systems, the Trusted Computer System Evaluation Criteria (the Orange book) [26], where the reference monitor is mentioned. The diagram in Figure 1 depicts the logical structure of the RM.

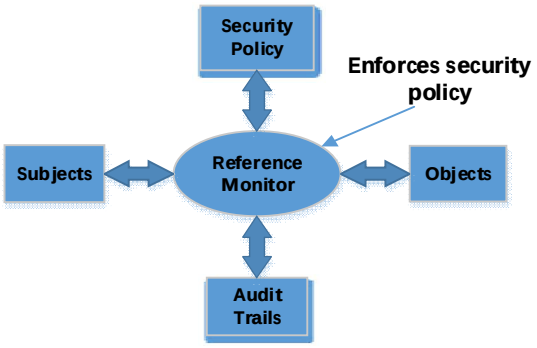


Figure 1. Logical Structure of RM

Implementations of the reference monitor concept make use of several traditional security measures, which apply to the agent environment. Such conventional techniques include cryptographic methods for authentication of AP/agent, encryption of data, integrity of data, and access control methods, thereby providing an integrated security solution. Thus, an implementation of the RM can be used to enforce security. The security policy for agent computing can involve different security requirements, however, any AP which implements an RM entity can be considered to be trustworthy. The mobile agent may then migrate to such an AP for secure execution. Thus, the presence of an RM entity which enforces security suffices to turn any remote and unknown host into a trusted computing base in an open system. In the case of mobile agent computing, we propose that the RM entity be in the form of an agent (RM-agent) which runs on the each AP. We define the following fundamental assumptions about the trust association of all entities in the security framework, for providing mobile agent security. These trust associations comprise the trust model in our system.

- The Certificate Authority (CA) authenticates principals and issues certificates to entities such as the agent owner, AP administrator, RM-agent, trusted-third party (TTP) RM entity distributor. Users/agent owners and APs with valid certificates are considered trustworthy.
- Trusted Third Parties (TTPs) are principals with genuine certificates from CA and are trustworthy. These TTPs distribute standard RM-agents, which satisfy some design specifications, for use by any AP. Such TTPs can also act as subordinate CAs. The trusted CA delegates them the right to issue certificates to APs. Then, the TTP can also maintain a directory server where it stores information about each AP and their corresponding RM-agent.
- RM-agents on APs with genuine certificates from the TTP are trustworthy. These certificates are signed by the TTP who must have full knowledge of the RM agent's behaviour and capabilities and thus all the possible consequences of its operation. The system administrator of the AP can request for an RM-agent and its certificate from the TTP,
- Agents signed by trustworthy owners are trustworthy. The agent-owner is expected to have knowledge of the capability and behaviour of the agent and to take responsibility of the actions of the agent acting on behalf of the user/owner.

The underlying idea of our trust model is based on the usage of undeniable proofs like digital signatures, i.e., RM-agents are signed by the TTP (using the private key of the TTP), thus ensuring that the RM-agents are genuinely from the TTP. The digital signature of the RM-agent also allows to check the integrity of the RM-agent, i.e., indication of tampering, if any, on the RM-agent. In the next section, we describe how the RM-agent can be delivered securely to an AP and how a mobile agent can assess the authenticity and integrity of the RM-agent before deciding to migrate to the remote AP. In this work, we focus on the RM integration into the AP to establish trust, rather than on how the RM-agent enforces the security policy to provide security to mobile agents executing on the AP.

IV. INTEGRATION AND EVALUATION OF RM-AGENT

In this section, we describe how the RM-Agent, which embodies the reference monitor concepts, can be securely obtained and integrated in an AP to establish trust so as a mobile agent can safely migrate and execute on the destination AP.

A. *Distribution and Integration of RM-agent into an AP*

The different steps required for an AP to obtain a trusted RM-agent from a TTP in a secure manner is illustrated in Figure 2.

Step 1: AP registers with TTP and requests an RM-agent for enforcing security policy. The security to be provided by the RM-agent, and thus the AP, may vary based on the security policy of the AP.

 ISBN: 978-1-61208-427-5

Step 2: TTP generates a digital certificate ($RM_{TTPCert}$) for the RM-agent. This certificate contains information that would allow the receiving AP to verify the authenticity and integrity of the received RM-agent from TTP. Some important information on the $RM_{TTPCert}$ certificate includes: identity of AP to whom RM-agent is distributed; cryptographic hash value of RM-agent code; digital signature of the RM-agent, version no. of RM-agent, and security policy ID implemented by RM-agent. The $RM_{TTPCert}$ certificate is itself signed by the TTP.

Step 3: The TTP sends the RM-agent and the $RM_{TTPCert}$ certificate to the AP administrator/owner.

Step 4: Upon receipt of the RM-agent code and $RM_{TTPCert}$ certificate from the TTP, the AP administrator/owner can verify that the signature of the certificate is correct. The AP administrator/owner then uses the digital signature of the RM-agent, provided in the $RM_{TTPCert}$ to ensure that RM-agent was sent by the TTP (authenticity check). Finally, the AP administrator/owner, checks if the hash of the RM-agent file downloaded is the same as that on the certificate to ensure that the RM-agent has not been replaced or tampered with (integrity check). AP administrator/owner also checks other information on the certificate.

Step 5: Finally, the AP administrator integrates the RM-agent in the AP to establish the AP as a trusted entity which provides security as per the security policy ID.

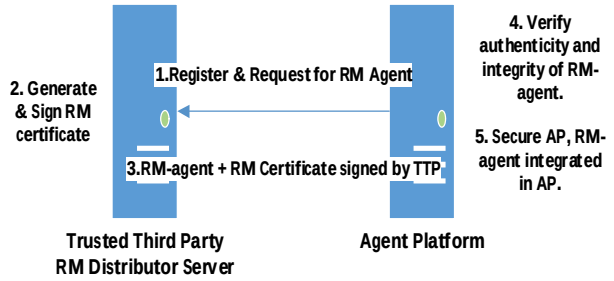


Figure 2. Distribution of RM-agent by TTP distributor

Thus, it can be seen that the RM-agent can be easily and securely downloaded from a TTP by any AP in an open system and integrated into the agent execution environment to provide trust enhanced security.

B. Verifying Authenticity and Integrity of RM-agent

Before a mobile agent migrates to the destination AP, if security is required, it has to ensure that RM-agent running on the destination AP has not been modified and is truly from a TTP. This problem is heightened by the fact that in an open system, the destination AP may not be known by the mobile agent. Verifying the trustworthiness of the RM-agent on a remote AP is necessary as one of the property of the RM is that it should be tamperproof and verifiable. It is true that the RM-agent has a certificate, provided by the source TTP ($RM_{TTPCert}$), but the certificate is no guarantee that the RM-agent running on the AP has not been modified. The $RM_{TTPCert}$ from the destination RM-agent allows the mobile agent to check the signature of the certificate to determine if the certificate and the data it contains are genuine and unmodified. The certificate shows that the RMagent was obtained from a TTP. The certificate also contains the security policy ID which informs the agent of the security provided by the destination AP. However, neither the $RM_{TTPCert}$ certificate nor the digital signature of the RM-agent allows to verify that the RM-agent *running* on the destination AP is the same as the one distributed by the TTP. Thus, it is not sufficient to confirm the trustworthiness of the RM-agent on the destination AP.

Alternatively, if the mobile agent could compare the hash value of the RM-agent code from the certificate with the hash code of the actual RM-agent to verify authenticity and integrity of the RM entity. However, given that the RM-agent is on the destination AP, the mobile agent on a source AP cannot calculate the hash value of the RM-agent. The mobile agent cannot also rely on the hash value received by a destination AP, as the destination AP may send a stored hash value of RM-agent, while it is running a modified version of the RM-agent. In such circumstances, the static validation approach fails to allow the mobile agent on a source AP to determine the integrity and authenticity of the RM-agent running on the destination AP prior to migration. Verifying the integrity and authenticity of code running at a remote destination is a challenging issue which we address as follows.

We identify the following requirements regarding the verification of the integrity and authenticity of the RM-agent running on the destination AP:

1. The source AP should be able to verify the integrity and authenticity of the RM-agent on the destination AP by storing minimal information about the RM-agent.
2. The integrity and authenticity checking mechanism has to stay secure even if the destination AP is malicious.
3. The communication bandwidth used during the verification process should be minimal, i.e., it should not involve the transfer of large amount of data.
4. The verification mechanism should be efficient in terms of computation.
5. It should be possible to run the verification several times, if desired, to ensure that integrity and authenticity of the RM-agent is maintained at all times. Static validation actually fails to satisfy this requirement because the hash code can be stored and sent back by the destination AP whenever required.

We propose a dynamic validation mechanism that satisfies the above conditions based on the challenge response approach. The mobile agent on a source AP can thus effectively verify the integrity and authenticity of the RM-agent on the destination AP that it wishes to migrate to.

C. Authentication and Integrity Checking Protocol (AICP)

The task of checking the integrity and authenticity of the RM-agent on the destination AP in our security framework lies upon the RM-agent on the source AP, since it is a service that may be requested by any mobile agent. If the mobile agents themselves were to perform this task, it would have added unnecessary burden on the mobile agent. The mobile agents thus remain lightweight and are programmed to only perform their tasks in the application. Accordingly, when a mobile agent wishes to migrate from its current AP to

 ISBN: 978-1-61208-427-5

another, it requests the RM-agent on the source AP to assess the security of the destination AP. Given that mobile agents can only interact with one another by communicating using an Agent Communication Language (i.e. an agent cannot invoke a method of another agent but it can request the destination agent for some processing), the proposed dynamic validation of authenticity and integrity of the RM-agent involves interaction which we refer to as the *Authentication and Integrity Checking protocol (AICP).*

This interaction begins with the source RM-agent requesting the $RM_{TTPCert}$ certificate from the destination RM-agent. From the certificate, the source RM-agent learns about the TTP from which the destination RM-agent has been acquired. Based on this knowledge, the source RM-agent can retrieve the distributed RM-agent files *from the TTP* (same files/codes are running on the destination AP as the destination RM-agent), so that the hash value of the RM-agent can be computed locally. It is assumed that the TTP makes available the RM-agent file for download for such verification process. It may be argued that downloading the RM-agent code for this purpose may require a large amount of bandwidth and storage space on the source AP. Nonetheless, our implementation of the RM-agent in JADE results in an RM-agent Java class file which is 17 KB in size. JADE is used as it is one of the most popular APs and it is Java-based; most of the existing APs are Java-based APs. Given the increasing bandwidth capacity and high speed of today's network, retrieving and caching the RM-agent for a short period of time should not pose any problem. Assuming there are ten different TTP sources with different implementations of the RM-agent, and that each RM-agent is of the size 20KB, for an AP to temporarily cache all the RM-agents, only 200KB (0.2 MB) of storage space is required. Thus the proposed protocol is storage efficient.

Next, the source RM-agent issues a challenge (random number -R) to the destination RM-agent. This challenge (R) can be sent encrypted using the public key of the destination RM-agent. The source RM-agent also calculates the expected response as follows and as depicted by Figure 3.

- Challenge (R) is concatenated with the RM-agent class file (RMCF): R || RMCF
- The concatenated output is hashed to obtain the hash value H(R || RCMF)

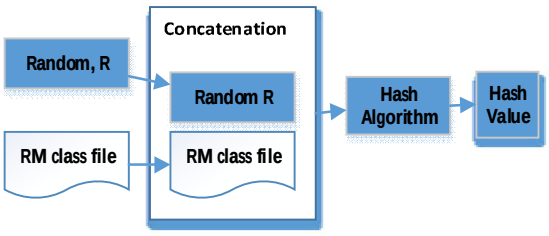


Figure 3. Calculating the response of the challenge at source

Similarly, the destination RM-agent uses its Private key to decrypt the encrypted Challenge sent by the source RM-agent. This step ensures that only the destination RM-agent has access to the Challenge (R). The destination RM-agent concatenates the Random number R to the current actual RM-agent file and then calculates the hash value of the concatenated input as shown in Figure 4. The response from the destination RM-agent is compared with the expected response computed by the source RM-agent. If they matches, then it is safe to assume that the destination RM-agent has not been tampered with (integrity). If the two hash values are the same, we can also assume authenticity of the destination RM-agent as the hash value has been computed using the RM-agent file from the TTP.

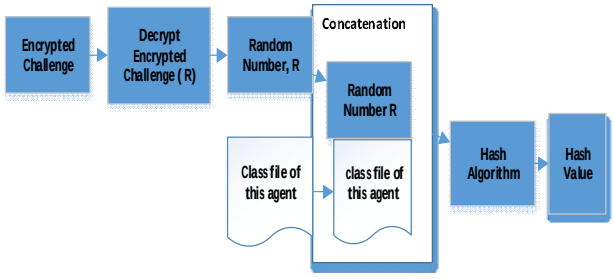


Figure 4. Calculating the response of the challenge at destination

V. EXPERIMENTS AND RESULTS

Experiments were designed with JADE agent development environment. The destination RM-agent uses its **own class file** for generating the Hash value. Using the this.getClass() method allows to find the agent's Java class filename. Once the class filename is known, the path of the file can be found. Because the RM-agent is running, the RM-agent java class file is read-only, i.e., cannot be modified. We read the file and copy it into another file, e.g., CopyFile. Concatenation is implemented by appending the Random number to the "CopyFile". The "CopyFile" is then hashed using SHA-256 to generate the hash value. Thus, it can be observed that the hash value generated truly correspond to the destination RM-agent. Similarly, on the source RM-agent, the RM-agent class file is read and copied into a new file, to which the random number is appended in the end (concatenation). This new file is passed as an argument to the hash function to generate the expected hash value. The same hash function is used by the source RM-agent and the destination RM-agent.

Given that in the context of agent computing all interaction takes place by means of ACL messaging between agents, the AICP is implemented as a series of messages exchanged between the source RM-agent and the destination RM-agent. The RM-agent is programmed by adding the following two behaviours: *InitiatorAICP* and *ResponseAICP* implemented as a OneShotBehaviour.

It was assumed that both the source AP and destination AP uses the same TTP source for the RM-agent. We run the AICP, with different versions of the destination RM-agent class file. It was observed that when the proper destination RM-agent was used at the destination AP, the response obtained was as expected. However, when the destination RM-agent code has been changed, the response obtained was different than the expected response. Thus, using the AICP, a source AP may successfully assess the authenticity and

 ISBN: 978-1-61208-427-5

integrity of the destination RM-agent and consequently the trustworthiness of the destination AP.

VI. CONCLUSIONS AND FUTURE WORK

We introduced the RM-agent, as a fundamental component of our framework. The RM-agent allows to implement the reference monitor concept, whereby a reference validation mechanism (implemented by the RM-agent), is responsible for enforcing the system's security policy. The RM-agent satisfies the requirements of the reference monitor concept in the sense that the mobile agent can verify that the RM-agent is from a trusted source (RM-agent is verifiable and trustworthy) and it is hasn't been tampered with. A novel dynamic verification approach was proposed for assessing authenticity and integrity of the remote RM-agent running on the destination AP.

RM-agents contribute to providing a secure computing environment to mobile agents. Altogether, with the security framework, it is possible for a mobile agent to migrate to any AP with the assurance of security. In this case, it is not important for a mobile agent to know every AP, as long as the AP deploys a RM-agent obtained from a TTP. Thus, our security framework is most suitable for open systems. Additionally, the security framework supports mobile agent computing with static as well as dynamic itineraries. Future works, revolves around equipping the RM-agent to ensure different security services by integrating different behaviours to the RM-agent. Then, the RM-agent will be able to enforce security such as code and data confidentiality, integrity, execution integrity.

REFERENCES

[1] R. R. Brooks, "Mobile Code Paradigms and Security Issues", IEEE Internet Computing, vol.8, no. 3, pp. 54-59, May/June 2004

[2] D. B. Lange and M. Oshima. "Seven good reasons for mobile agents." Communications of ACM , March 1999, 88-89.

[3] R. K Verma and S. Jangra, S., "Significance of Mobile Agent in Wireless Sensor Network". International Journal of Advance Research in Computer Science and Management Studies, 1(7), 2013, pp. 328-335.

[4] T.T. Khose Patil, and C, Banchhor, "Distributed Intrusion Detection System using mobile agent in LAN Environment." International Journal of Advanced Research in Computer and Communication Engineering, 2(4), 2013. pp. 1901-1903.

[5] V. Upadhyay, J. Balwan, G. Shankar, and Amritpal, "A Security Approach for Mobile Agent Based Crawler." Proceedings of the Second International Conference on Computer Science, Engineering & Applications (ICCSEA 2012) New Delhi, India, 2012.

[6] Y. Essa, G. Attiya, and A. El-Sayed, "Mobile Agent Based New Framework for Improving Big Data Analysis." IEEE International Conference on Cloud Computing and Big Data (CloudCom-Asia). Fuzhou, 2013.

[7] W. Godfrey, S. Jha, and S. Nair, "On a Mobile Agent Framework for an Internet of Things". IEEE International Conference on Communication Systems and Network Technologies (CSNT), 2013

[8] M. Higashino et al., "Management of Streaming Multimedia Content using Mobile Agent Technology on pure P2P-based Distributed e-Learning System. Barcelona", In the Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications, 2013

[9] W. Jansen, and T, Karygiannis, " NIST Special Publication 800-19-Mobile Agent Security", Technical paper, Computer Security Division: National Institute of Standards and Technology, 2000.

[10] C. Lin, V. Varadharajan, "Trust Enhanced Security - A New Philosophy for Secure Collaboration of Mobile Agents", COLCOM, 2006, International Conference on Collaborative Computing: Networking, Applications and Worksharing, International Conference on Collaborative Computing: Networking, Applications and Worksharing 2006, pp. 76.

[11] Jensen, C. Damsgaard. "The Importance of Trust in Computer Security." Trust Management VIII. Springer Berlin Heidelberg, 2014. pp 1-12.

[12] X. Guan, Y. Yang, and J. You, "POM-A mobile agent security model against malicious hosts". IEEE Fourth International Conference on High Performance Computing in Asia-Pacific Region. Beijing, China, 2000

[13] S. Zhidong, and T. Qiang, "A Security Technology for Mobile Agent System Improved by Trusted Computing Platform." Proceedings of the Ninth International Conference on Hybrid Intelligent Systems (HIS 2009) , Shenyang, 2009.

[14] S. Loureiro, and R. Molva, "Mobile code protection with smartcards." Proceedings of the Sixth ECOOP Workshop on Mobile Object Systems: Operating System Support, Security and Programming Languages, Cannes, France,, 2000.

[15] S. Armoogum, and A. Caully, "Obfuscation Techniques for Mobile Agent code confidentiality." Journal of Information & Systems Management, 1(1), 2011 pp. 25-36.

[16] B. Barak, et al. "On the (im)possibility of obfuscating programs." Proceedings of the 21st Annual International Cryptology Conference - Advances in Cryptology, Santa Barbara, California, USA, 2001, pp. pp 1-18.

[17] O. Esparza, M. Soriano, J. L. Munoz, and J. Forne, "A protocol for detecting malicious hosts based on limiting the execution time of mobile agents." IEEE Eight International Symposium on Computers and Communication (ISCC'03), Kemer-Antalya, Turkey, 2003

[18] T. Sander, and C. F. Tschudin, "Towards Mobile Cryptography." Proceedings of IEEE Symposium on Security and Privacy. Oakland, CA, 1998

[19] H. Lee, J. Alves-Foss, and S. Harrison, "The use of Encrypted Functions for Mobile Agent Security." IEE 37th International Conference on System Sciences, Hawaii, 2004

[20] U. G. Wilhelm, S. Staamann, and L. Buttyn. "On the problem of trust in mobile agent systems." In Proceedings of Network and Distributed Security Symposium, San Diego, California, Internet Society, 1998

[21] C. Lin, V. Varadharajan, "MobileTrust: a trust enhanced security architecture for mobile agent systems", International Journal of Information Security, Volume 9, Issue 3, 2010, pp 153-178

[22] D. Foster, V. Varadharajan, " Security and trust enhanced mobile agent based system design," Third International Conference on Information Technology and Applications. 2005, Vol 1 pp 155 - 160

[23] J. Anderson, " Computer Security Technology Planning Study, Bedford MA: Technical Report ESD-TR-73-51", Air Force Electronic Systems Division, Hanscom AFB. 1972

[24] Jaeger, T., 2011. Reference Monitor. Encyclopedia of Cryptography and Security (2nd Ed.), 2(1), pp. 1038-1040.

[25] C. E. Irvine, "The Reference Monitor Concept as a Unifying Principle in Computer Security Education." Proceedings of the International Federation for Information Processing (IFIP 99) & 1st World Conference on Information Security Education. Kista, Sweden, 1999

[26] D. o. D., "Trusted Computer System Evaluation Criteria, Orange Book", Library No. S225 711: Dept of Defense CSC-STD-001-83. 1983

 ISBN: 978-1-61208-427-5

An IDS for Browser Hijacking

Diogo Mónica and Carlos Ribeiro

INESC-ID/IST
Instituto Superior Técnico
Lisboa, Portugal
Email: diogo.monica@ist.utl.pt, carlos.ribeiro@ist.utl.pt

***Abstract*—The steady evolution of browser tools and scripting languages has created a new, emergent threat to safe network operations: browser hijacking. In this type of attack, the user is not infected with regular malware but, while connected to a malicious or compromised website, front end languages such as javascript allow the user's browser to perform malicious activities; in fact, attackers usually operate within the scope of actions that a browser is expected to execute. Paradigmatic examples are the recent attacks on GitHub, where malicious javascript was injected into the browser of users accessing the search-giant Baidu, launching a devastating denial-of-service against a US-based company. Detecting this type of threat is particularly challenging, since the behavior of a browser is context specific. Detection can still be achieved, but to effectively hamper the effectiveness of this type of attack, users have to be empowered with appropriate detection tools, giving them the ability to autonomously detect and terminate suspicious types of browser behavior. This paper proposes such a tool. It uses information available within the browser (and is, thus, implementable as a browser extension), and it allows users to detect and terminate the suspicious types of behavior typical of hijacked browsers.**

Keywords–IDS; Browser hijacking; Malicious attack detection; User empowerment.

I. INTRODUCTION

The detection of malicious behavior that does not directly target the user's browser is largely unexplored. Since the majority of existing detection methods focus on the detection of specific attacks (exploit attempts, for example), behaviors such as having javascript code send thousands of back-to-back connections to different remote destinations is typically not considered malicious, even though it constitutes a behavior associated with several types of attacks, from browser-based denial-of-service [1][2], stealth botnet command-and-control [3] and even network reconnaissance techniques, such as network and port scanning [4]. Another example of undesired behavior that a malicious javascript might inflict on a browser is the use of the host's computational power to, for example, participate in mining crypto-currencies [5]. Unfortunately, until recently, browsers did not provide access to statistics on how a particular tab was behaving, giving the user information only on behavior of the browser as a whole. By leveraging the fact that browsers such as Google Chrome now allow the access to very granular information about the actions being taken by the browser itself, we can create tools that analyse the behavior of each tab individually, and inform the user of potentially suspicious behavior.

In this article, we will show results which indicate that, with the appropriate choice of the classification dimensions, a linear multi parameter detector is capable of flagging such attacks and, hence, of giving users the capability to disconnect from malicious sites if and when their browser is being wrongly used. In fact, empowering users with such a detector allows us to go further ahead in the road of cooperation between users and researchers. One of the problems that are yet to be solved, is the fact that even for samples of javascript that can be detected and classified as malicious by existing tools, we still have to come in contact with them in the first place. Crawling the internet seems like a losing battle; it is just not feasible to do it at a rate where a piece of malicious code can be detected before the attacker has already taken advantage of the bandwidth and processing power of several thousands of users. If users can detect the hijacking of their browsers for malicious purposes, then it will be possible to have these users' browsers automatically submit URLs where potentially malicious javascript might be running, in an anonymous fashion. This, together with the unification of the several blacklists currently in existence, could be a great defuser (and deterrent) of this kind of attack.

The vulnerabilities of front end languages such as javascript have long been recognised, and much work has been devoted to protection against the risks posed by the considerable capabilities of such languages to the user's systems (e.g., [6][7][8]). These approaches typically limit the power of javascript, either by restricting the language to an accepted subset of capabilities, or simply by monitoring the scripts degree of adherence to a designated secure framework, and dynamically modifying untrusted code to allow only operations and API calls deemed to be secure. The major drawback of such an approach lies in the fact that it effectively hampers front end capabilities for both good and evil purposes alike. The use of data driven approaches to analyse network traffic behavior and, hence, build intrusion detection systems (IDS) by anomaly detection, has also received much attention; a plethora of tools have been proposed, ranging from purely stochastic approaches, to machine learning based clustering and classification algorithms (e.g., [9][10][11][12][13][14]). These approaches are network oriented, and are typically based on network wide routing and/or protocol execution data.

Hence, most of the tools used in this paper have already been individually used and proposed in the general field of IDS, even though in different particular contexts, and with different particular objectives. To the authors knowledge, the approach proposed in this paper is, however, unique, due to the set of properties it presents: i) isolated operation: detection is achieved by individual users, without the need

 ISBN: 978-1-61208-427-5

for any network related data; ii) No cooperation from the site hosting the script is required; iii) The set of capabilities of the scripting language is not restricted in any way; iv) the decision parameters required by the classifier are all obtainable within the user browser, thus allowing for algorithm implementation as a simple browser extension. Hence, this approach empowers users in a very effective and simple way, allowing them to autonomously detect the hijacking of their browsers for malicious purposes. As previously discussed, this will heavily contribute to safer network operation. The rest of this paper is organized as follows: Section II describes the detector, its rationale and implementation. Section III discusses the obtained experimental results. Section IV concludes the paper.

II. DETECTION

Even though general in scope and nature, the proposed tool is currently focused on providing an effective defense against the type of threats that use legitimate users' browsers to amplify the network capabilities of the attackers. Good examples are the javascript version of Low Orbit Ion Cannon (JS LOIC) [15], a denial-of-service (DoS) tool used by the hacker group Anonymous, and the javascript based botnet command-and-control discussed in [3]. As such, the detection problem setup will be based on the type of behavior adopted by a hijacked browser in these particular cases.

While the objective of a malicious payload that delivers a DoS tool is fundamentally different than a payload that allows attackers to do network reconnaissance, both imply a dramatic increase in the rate of new HTTP requests, and a subsequent increase in CPU usage of the browser tab that hosts the malicious code. In the particular case of the command-and-control architecture described in [3], there is an initial phase of random (or semi-random) IP scanning, where hijacked browsers are used in an attempt to disseminate malicious commands to one or more infected bots. Due to this massive scanning phase, any recruited browsers will dramatically increase the rate of new HTTP requests to distinct destination IP addresses. If we are able to reliably detect either this bot reaching behavior, or the less complex situation of a DoS based on HTTP requests flooding, the browser can automatically terminate the session with the site, or simply close the affected tab, effectively stopping the attack. Also, as previously mentioned, it will be possible at this point, to automatically submit URLs where potentially malicious javascript might be running, something which could be a great defuser of this kind of attack.

The proposed detector will thus be designed to detect this scanning and request flood behavior. No single parameter can, however, constitute a reliable indicator of this malicious behavior. An increase in HTTP requests may be the result of a legitimate action; an increase in the computational effort being consumed by the browser may easily occur in many legitimate instances (e.g., video streaming); in general, the same can be said of any single parameter/indicator. We will, thus, employ a multiparameter detector. Each one of the used dimensions will produce an indicator which, in itself, will be incapable of completely typifying the "hijacked" behavior but, taken together, they will be capable of flagging this behavior, as will be seen.

The associated detection problem is, however, not trivial. Since a false alarm will imply the disconnection from the site (or the tab) and, thus, a serious inconvenience to the user, it is important to guarantee the adequacy of the detection algorithm. Ad-hoc heuristics based on a single parameter evaluation of, for example, new HTTP requests to new IPs (mean rate, maximum rate, effective rate, accumulated number of requests, etc) are prone to fail, and are easily deceivable; multi-parameter heuristics (e.g., a linear combination of the above parameters) are more complex, they are typically highly arbitrary in the parameter weights, don't always generate an adequate final metric, and are not easily scalable, due to the many dimensions along which scaling can be independently done. The problem of multi-criterion detection will therefore be handled by a data driven mechanism, to avoid imposing arbitrary heuristic rules to the detector. In this article, the classifier will be a simple perceptron, but any equivalent linear classifier might have been used. It will be trained with different instances of "normal" and "hijacked" browser behavior. Being a linear classifier, the performance of the perceptron will depend on the two classes of behavior ("normal" and "hijacked") being, or not being, linearly separable [16]. As will be shown below, in all performed tests, the rate of correct classification was 100%, which implies that, with the chosen classification dimensions, the problem seems to be, indeed, linearly separable.

The proposed detector should be implementable as a browser extension and will, thus, be based solely on the data accessible by the browser, concerning its own behavior. It will be based on three different dimensions (all of them accessible and quantifiable within the browser itself, on a per-tab basis):

- Computational effort;
- Periodicity of new HTTP requests;
- The sequence of destination IP addresses of new HTTP requests;

The rationale behind the first indicator (computational effort) is clear. An attempt to establish an effective massive scanning strategy will necessarily correlate with an increase in the computational effort required by that browser's tab. The same is true if the browser is recruited to perform a DoS attack.

The second indicator has a more subtle rationale: independently of the computational power of the host and, therefore, of its achievable maximum rate of new HTTP requests, whenever the host is driven close to its maximum power in the massive task of flooding a single remote target or scanning the full internet address space, the new HTTP requests will become increasingly periodic, the period being dictated by the minimum cycle time achievable with the host's computational abilities; it is, thus, a dimension which, even though capable of indicating a browser hijack, is fairly insensitive to the amount of computational power available to the host computer.

The third and last indicator (sequence of destination IP addresses of new HTTP requests) is used to capture the addressing schemes typical of blind scanning strategies.

As has already been stated, none of these indicators will be capable of completely typifying the "highjacked" behavior, since periodicities will arise in legitimate video streaming, high computational loads will appear in many legitimate instances, etc; the same can be said, *mutatis mutandis*, for each one of the individual indicators. However, when considered together, they will be capable of flagging this behavior, as will be seen below.

Obtaining the indicators to feed the perceptron will require some pre-processing of the browser traffic data in each one of

 ISBN: 978-1-61208-427-5

the considered dimensions, as will be seen. A fourth parameter (absolute number of new HTTP requests during the analysis period) will be used, not as a decision parameter, but as an enabler mask. This will be detailed below.

A. Computational effort

This dimension of the detection scheme evaluates the fraction of available computational effort that a particular browser tab is requiring. In a sense, this is the easiest one of the three indicators to be obtained, since we can obtain the fraction of the available computing power being consumed by a tab, directly from the browser. To account for the possibility of different profiles in the computational requirements within the analysis period, the obtained data passes through an integrator finite impulse response (FIR) filter. Hence, only the total computational power consumed in the analysis period (as a fraction of the available computational power in that period) is considered. Every second, a new measure of the fraction of computational power being consumed by the browser's tab is taken ($c(n)$, n being the time index of the sample). This sample is fed to an integrator FIR filter, which implements the composite Trapezoidal Quadrature rule, whose output p_n is the total consumed computational power over the last 5 seconds; the Trapezoidal rule was chosen due to its more acceptable behavior at sampling rates close to Nyquist, when compared with other short impulse response rules such as, for example, the Simpson rule (see, for example, [17], for details.).

The filter's gain is also adjusted, to guarantee that its output remains in the interval [0,1]. Since the interval between samples is $1\,s$, the filter's impulse response has 6 taps; the integrator filter thus becomes (noting that $0 < c(i) < 1, \ \forall\, i$):

$$p_n = \frac{1}{10}\left(c(n) + 2\sum_{i=1}^{4} c(n-i) + c(n-5)\right)$$

Since this filter is implemented as a sliding window, we have

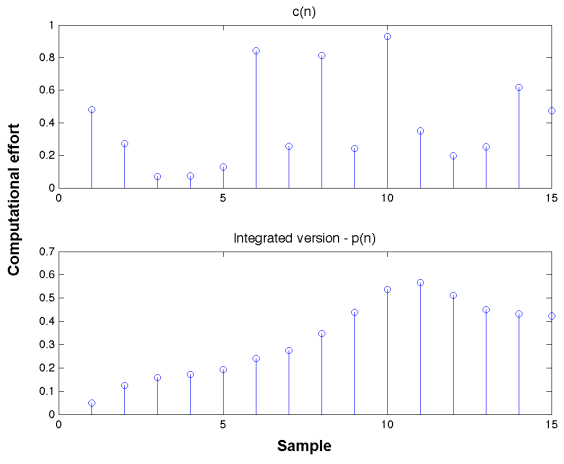


Figure 1. Computational indicator.

a new value of p_n at every second (see Figure 1), to be used as an indicator of the "computational effort" dimension, and, therefore, to be directly fed into the detector. This indicator is, therefore, valued in the $[0, 1]$ range, with higher values corresponding to higher computational loads.

B. Periodicity of new HTTP requests

As discussed, whenever the host is driven close to its maximum power in the single massive task of scanning the internet address space, the new HTTP requests will become increasingly periodic. The period depends on the host's computational abilities and the cycle time it can achieve, but the appearance of a periodic behavior may itself be used as an indicator. To decrease the computational complexity of the periodicity evaluation, no spectral analysis is performed. The indicator is, thus, obtained strictly via time domain processing, and relies on the analysis of the time between successive new HTTP requests.

If new connection requests appear in a purely random fashion, a Poisson arrival process is to be expected. This means that the times between successive connection requests (inter-arrival times) should be exponentially distributed (see, e.g., [18]). One useful measure of the type of use being required from the browser can, therefore, be obtained by testing the sequence of inter-arrival times, and determining if their distribution is, indeed, exponential (implying that we are facing random, non periodic, requests), or if they have some deviation from this pure theoretical random pattern. A simple Kolmogorov-Smirnoff (KS) test may be used for this purpose (e.g., [19][20]). The test proceeds as folows [19]:

As is standard in the KS test, it relies on comparing the sample cumulative distribution function with the cumulative exponential distribution function: given a sequence of N observations, one determines

$$d = max\left\{\left|S(X) - C(X)\right|\right\}, \tag{1}$$

where $S(X)$ is the sample cumulative distribution function, and $C(X)$ is a cumulative exponential distribution function with mean $\bar{\mu}$, the sample mean. If d exceeds a given threshold Ψ, one rejects the hypothesis that the observations were taken from an exponential distributed population. For sequences of 30 or more samples, and a level of significance of 0.05, the threshold Ψ may be approximated by [19]:

$$\Psi = \frac{1.06}{\sqrt{N}}, \tag{2}$$

N being the number of samples in the sequence. For lower values of N, the corresponding values of Ψ can be found in [19].

We can, therefore, easily verify if the inter-arrival times are (or are not) exponentially distributed. However, "not being exponential" is far from being a sufficient indicator that an attack is under way. As such, instead of using a binary variable to represent the exponential/not exponential nature of the inter arrival times as in the KS test, we will use the (continuous) ratio d/Ψ. Also, a complementary measure will be used: the ratio $\sqrt{\sigma^2}/\mu$ of the inter arrival times (σ^2 being their variance, and μ their mean value); when the HTTP requests become increasingly periodic, this ratio becomes increasingly smaller (since the inter-arrival times become increasingly concentrated around the mean) and, as such, it constitutes an additional measure of periodicity. To compute this ratio at each moment (the sampling period is again 1 second), we consider the inter arrival-times observed in the previous 5 seconds. The estimate for the mean arrival time is the sample mean ($\bar{\mu}$), and the

 ISBN: 978-1-61208-427-5

estimate for the variance is obtained by the unbiased sample variance:

$$\bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (3)$$

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{\mu})^2 \quad (4)$$

(5)

Combining the two previous measures, the final indicator for periodicity of new HTTP requests becomes:

$$\text{per}_n = \frac{d\sqrt{s^2}}{3\Psi\bar{\mu}}. \quad (6)$$

The scaling factor $1/3$ is used simply to maintain a common scale to all tree used indicators.

C. The sequence of IP addresses of new HTTP requests

The behavior of a hijacked browser in what concerns the pattern of contacted IP addresses varies widely with the type of attack into which the browser is recruited. While attacks such as DoS will result in an endless repetition of a single IP (or a short list of IPs), an attack where the hijacked browser must scan the internet (as in the case of the referred stealth botnets) will generate a very high dispersion of addresses where, typically, no address is attempted more than once. To further complicate matters, the strategies for blind scanning may also differ widely, ranging from deterministic sequential scanning to purely random scanning. To address the issue with a single indicator, we used a two step procedure, applied, at each second, to the list of IP addresses of new connections attempted in the 5 s period ending at moment n ($l_n(i),\ 1 < i < M$), M being the number of new connections attempted within that period :

1) A new list L_n is obtained, by computing the absolute values of the second order divided diferences of l_n:

$$L_n = |\,\Delta(\Delta(l_n))\,|, \quad (7)$$

where the Δ operator is defined by:

$$\Delta(l_n) = \{l_n(i) - l_n(i-1),\ 2 < i < M\}; \quad (8)$$

2) If all elements of L_n are 0, indicator addr_n is considered to be 0. Otherwise, $\text{addr}_n = \sigma_{L_n}/3\mu_{L_n}$, where μ_{L_n} is the mean value of elements in L_n, σ_{L_n} its standard deviation, and the $1/3$ factor is, again, simply a scaling factor. That is:

$$\text{addr}_n = \begin{cases} 0 & \text{if } L_n(i) = 0,\ \ 1 < i < M-2 \\ \frac{\sigma_{L_n}}{3\,\mu_{L_n}} & \text{otherwise} \end{cases} \quad (9)$$

This indicator thus has the following properties: i) DoS attacks to a single address will map to $\text{addr}_n = 0$ (due to the inner Δ operator); ii) Sequential scanning strategies will map to $\text{addr}_n = 0$ (due to the sequence of inner and outer Δ operators); iii) random scanning strategies will also map to low values of addr_n, due to the resulting high standard deviation of L_n. This means that both DOS attacks and the two most common scanning strategies (sequential and random) will generate low values of this indicator, something which makes it a powerful dimension of the detection scheme.

D. Absolute number of new HTTP requests

As will be seen, with the previous three indicators, the perceptron is capable of separating the "normal" and "hi-jacked" cases, for all periods with a reasonable amount of activity. However, in some periods when the browser is idle, the indicators are incapable of characterizing the activity, due to an insufficient number of new connections and, thus, the unreliabilty of the derived statistics. To account for those cases, and since the absence of traffic is a consistent indicator that no attack is under way, a threshold $\Phi = 10$ is established for the minimum number of new HTTP requests, under which it is always assumed that no attack is under way. In fact, with such low rates of probing, and the correspondingly high intervals between probes, any eventually ongoing attack (be it a DOS attack, or an attempt to scan the internet address space) would be highly ineffective, and thus inexistent as a real threat, in this context.

III. Experimental results

To train the perception performance, 50 browser sessions were logged. From these sessions, 450 non-overlapping 5 seconds periods were extracted, to be used as training set (D); of these, 150 correspond to regular browser use, 150 to a simulated DOS attack, and 150 to forced random scanning periods; 50 other periods were obtained, to be used as a test set. The three indicators x_1, x_2, and x_3 for the training 450 periods were fed to the perceptron, for supervised training; 100 iterations (epochs) were used in training, with a learning factor $\alpha = 0.1$; the perceptron weights $\mathbf{w}$ were randomly initialized. At each epoch, all samples were presented to the perceptron, sequentially, in random order. For each sample $\mathbf{x}(i) = [1, x_1(i), x_2(i), x_3(i)]$ in the training set D (the indicator vectors have been extended with a trailing 1, for mathematical convenience in the training equations below), training proceeds as follows (see, e.g., [21][22], for further details on the perceptron concept, design and training):

1) Obtain the perceptron output:

$$y(i) = f\Big(\mathbf{w}(i)\cdot\mathbf{x}(i)\Big) = f\Big(w_0(i) + \quad (10)$$
$$+ \quad w_1(i)x_1(i) + w_2(i)x_2(i) + w_3(i)x_3(i)\Big),$$

where

$$f(\tau) = \begin{cases} 1 & \text{if } \tau > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

2) Update the weights:

$$\mathbf{w}(i+1) = \mathbf{w}(i) \ + \alpha(d(i) - y(i))\mathbf{x}(i), \quad (12)$$

$d(i)$ being the correct decision label (1- attack, 0 - no attack) for the i^{th} 5 second period.

After training, classification of a given observed period is done simply by computing

$$y(i) = f\Big(\mathbf{w}(i)\cdot\mathbf{x}(i)\Big) = f\Big(w_0(i) + \quad (13)$$
$$+ \quad w_1(i)x_1(i) + w_2(i)x_2(i) + w_3(i)x_3(i)\Big),$$

with $\mathbf{w}$ being the final set of weights that resulted from the training phase, and $\mathbf{x}(i) = [1, x_1(i), x_2(i), x_3(i)]$ the augmented indicator vector for that period.

 ISBN: 978-1-61208-427-5

When the 450 training periods were run through the trained perceptron, no errors in classification were observed, which means that the problem, as mapped by these three indicators, is indeed linearly separable, within the training set. For the test set (constituted by 50 5-seconds period, as discussed above), classification was also 100% successful, with no misclassifications. Even though the number of examples used in this paper is somewhat limited, and no real life attacks have been used, the obtained results seem to indicate that the proposed indicators may, indeed, be capable of transforming the browser highjacking detection problem into a linearly separable one, thus addressable by simple linear classifiers, implementable by simple, lighweight, browser extensions.

IV. CONCLUSION

Our results show that, with the appropriate choice of indicators, it seems to be possible to create a linearly separable setup, amenable to the detection of browser hijacking by malicious sites with a simple linear detector. This conclusion must still be validated with bigger, real life, datasets. Detection is accomplished using only variables which the browser provides access to, and it can be done with a per-tab granularity. Two main types of attack are thus defused: attacks that utilize honest user's browsers as a platform to launch denial-of-service, and attacks which imply mass IP address scanning (both sequential and random). In particular, this detector allows users to become a vital part in defusing a particularly dangerous type of stealth botnet, by detecting that their browser is being used as part of the botnet command-and-control structure. Finally, our solution also paves the way for automatic submission of URLs where potentially malicious javascript might be running, something which can be a great defuser and deterrent for future attacks.

REFERENCES

[1] J. Temperton, 2015, [Accessed 11 July 2015]. [Online]. Available: http://www.wired.co.uk/news/archive/2015-04/10/china-great-cannon-github-hack

[2] K. Higgins, 2013, [Accessed 11 July 2015]. [Online]. Available: http://www.darkreading.com/attacks-breaches/ddos-attack-used-headless-browsers-in-150-hour-siege/d/d-id/1140696

[3] D. Oliveira and C. Ribeiro, "Leveraging honest users: Stealth command-and-control of botnets," in Proceedings of the 7^{th} USENIX Workshop on Offensive Technologies (WOOT '13), Washington, D.C., August 2013.

[4] L. Kuppan, 2010, [Accessed 11 July 2015]. [Online]. Available: http://www.andlabs.org/tools/jsrecon.html

[5] R. King, 2013, [Accessed 11 July 2015]. [Online]. Available: http://qz.com/154877/by-reading-this-page-you-are-mining-bitcoins/

[6] P. Phung, M. Monshizadeh, M. Sridhar, K. W. Hamlen, and V. Venkatakrishnan, "Between worlds: Securing mixed javascript/actionscript multi-party web content." in IEEE Trans. Dependable and Secure Computing, 2014.

[7] P. Phung, D. sands, and A. Chudnov, "Lightweight self-protecting javascript," in Proceedings of ASIACCS09, March 2009.

[8] Y. Dachuan, A. Chander, N. Islam, and I. Serikov, "Javascript instrumentation for browser security," in Martin Hofmann 0001 Matthias Felleisen, ed., POPL, ACM, 2007, pp. 237–249.

[9] H. Kayacik, A. Zincir-Heywood, and M. Heywood, "On the capability of an som based intrusion detection system," in Proc. Int. Joint Conf. Neural Networks, vol. 3, July 2003, pp. 1808–1813.

[10] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, and M. Embrechts, "Network-based intrusion detection using neural networks," in Proc. Artif. Neural Netw. Eng., vol. 12, November 2002, pp. 579–584.

[11] P. Kabiri and A. Ghorbani, "Research on intrusion detection and response: A survey." in International Journal of Network Security, vol. 1(2), 2005, pp. 84–102.

[12] J. Cabrera, B. Ravichandran, and R. Mehra, "Statistical traffic modeling for network intrusion detection," in Proc. Modeling, Anal. Simul. Comput. Telecommun. Syst., 2000, pp. 466–473.

[13] D. Denning, "An intrusion detection model," in IEEE Trans. Softw. Eng., vol. SE-13, N.2, February 1987, pp. 222–232.

[14] S. Jiang, X. Song, H. Wang, J. Han, and Q. Li, "A clustering-based method for unsupervised intrusion detections," in Pattern Recognition Letters, Elsevier, vol. 27, 2006, pp. 802–210.

[15] P. Shankdhar, 2013, [Accessed 11 July 2015]. [Online]. Available: http://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools/

[16] F. Rosenblatt, "The perceptron–a perceiving and recognizing automaton," in Report 85-460-1, Cornell Aeronautical Laboratory, Jan 1957.

[17] R. Hamming, Digital filters, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1983.

[18] A. Papoulis, Probability, Random Variables and Stochastic Processes, 2nd ed. New York, USA: McGraw-Hill, 1984.

[19] H. W. Lilliefors, "On the kolmogorov-smirnov test for the exponential distribution with mean unknown," in Journal of the American Statistical Association, vol. 64, March 1969, pp. 387–389.

[20] V. Seshadri, M. Csorgo, and M. A. Stephens, "Tests for the exponential distribution using kolmogorov-type statistics," in Journal of the Royal Statistical Society. Series B (Methodological), vol. 31, 1969, pp. 499–509.

[21] C. M. Bishop, Neural Networks for Paitern Recognition. Oxford: Clarendon Press, 1995.

[22] R. O. Duda and e. a. P. E. Halt, Pattern Classification. Wiley-lnterscience, John Wiley and Sons, 2001.

 ISBN: 978-1-61208-427-5

Monitoring of Malware Communication Channels

Radovan Holík, Roman Jašek
Department of Informatics and Artificial Intelligence
Tomas Bata University in Zlín
Zlín, Czech Republic
{rholik, jasek}@fai.utb.cz

***Abstract*—One of the trends in the security world of the 21st century has been a massive growth in malware. Anti-virus vendors make efforts to respond to the malware growth with constant development of anti-virus software and its updating signatures. In spite of this fact, there is a chance that even secured systems may be infected. Analysis of malware of Command and Control (C&C) servers is a technique for detecting unknown malware in anti-virus software. It allows for detailed understanding of the important aspects of malware and plays a key part in any forensic analysis. This paper is an initial work for future research and describes possible usage of this technique for a malware detection.**

Keywords-HTTP; DNS; C&C; malware detection

I. INTRODUCTION

The amount of malware has rapidly increased since the beginning of the twenty-first century, especially when compared to the end of the twentieth century [1][13]. In the years preceding the twenty-first century, malware used to be created to experiment with systems and also for authors of such malicious programs to rise to fame. Modern malware, however, is built for one purpose only – to make money. In particular, it aims at stealing and re-selling personal or company's data (e.g., user names, passwords, etc.) or resources (e.g., for botnets, etc.)[2][14]. In terms of their development, advanced techniques and methods are used to conceal attacks against modern security systems and anti-viruses, even though the general acceptance in Information Technology security is that all systems are bound to once fail. When a computer system is successfully attacked, it is necessary to prepare it for system recovery, screen it for a range of penetration threats, identify weak points in the system which enabled the penetration, and to provide steps to prevent similar penetrations in the future [15]. One of the possible ways to detect malware attacks is to monitor the malware communication channels. In the following, several basic methods for analyzing these channels are described. It is demonstrated that, although it is possible to get a lot of interesting information about the malware behavior in the network, the performance of the analysis does not require an advance knowledge of a systems analyst. Input data are easy to collect and many organizations follow this standard practice (e.g., for analyzing the network traffic). These are primarily records of domain names translation by Domain Name Server (DNS) protocol and records of computer accesses from a local network to Uniform Resource Locator (URL) in Internet (including heads of Hypertext Transfer Protocol (HTTP)). These records may be collected on clients and also on systems for a network monitoring.

The rest of the paper is organized as follows: command & control channels are described in Section 2; common obfuscation techniques are presented in Section 3; and Section 4 presents web services for analysis. Methods of C&C channels are discussed in Section 5, and conclusion will wrap up the paper.

II. COMMAND AND CONTROL CHANNELS

The analysis of communication channels of malware is inestimable in big organizations to detect penetrations. In small organizations, it is recommended that analysts respond when such incidents are discovered. Especially, it is important to focus on the inspection of the range of penetration and execution of the attack. In the following subsections, we focus in particular on analysis of HTTP and DNS protocols because these protocols are largely used by modern malware.

Monitoring of malware communication channels is a technique capable of revealing successfully attacked computers communicating to the world. This method is also effective if these computers have already been infected by malware for which no anti-virus signatures have been created. The analysis contains monitoring of used protocols, communicating sides and transmitted data. Variability of communication channels is lower than the variability of polymorphic malware, which allows grouping of malware into related families. In the case of targeted attacks, it also allows linking attacks between different organizations. Identification, blocking and disconnection of control servers are important weapon against malware worldwide as well. In certain cases, the monitoring of communication channels may be used also for detection of a range of targeted attacks. If an attacker is able to successfully crack a protected network, his next steps within the network are often done by already obtained credentials of ordinary users; therefore, without further use of malware. By identifying the primary communication channel and subsequent searching similar links, it is possible to detect an effort of the attacker using obtained data from the protected area [10].

 ISBN: 978-1-61208-427-5

The role of control servers traditionally perform computers controlled by malware. These computers have been successfully attacked during a previous attack. They are usually located in the same country as target of the attack (see Figure 1), in order to decrease the probability of successful detection. It is very difficult to track the people that perpetrate attacks, as they communicate with control server through several links distributed worldwide.

At the same time, the amount of the malware that resort to administration legitimate Internet services, such as network storage (Google Disk, One Drive, DropBox, etc.), social networks (Twitter, Facebook) and/or discussion forums, increases. For instance, in 2009 was detected Trojan.Grups, which used published messages on Google Groups for command distribution towards to control machines [4]. Similarly, Flashback malware receives commands via Twitter messages that contain specific hashtags [5], whereas network storage may be used as transship point for stolen data that the attacker tries to send quietly out of the targeted network. Moreover, network storage might be used for malware updates on infected computers [6]. The motivation for the usage of existing users' infrastructure is hiding in common data traffic and thus avoid detection mechanisms based for example on monitoring of communication with other than predefined servers. Basically, network services themselves decrease the effectiveness of detecting mechanisms. For instance, systems based on assigning a reputation to each Internet Protocol (IP) address have a limited use for cloud services with a risk of high number of false positives.

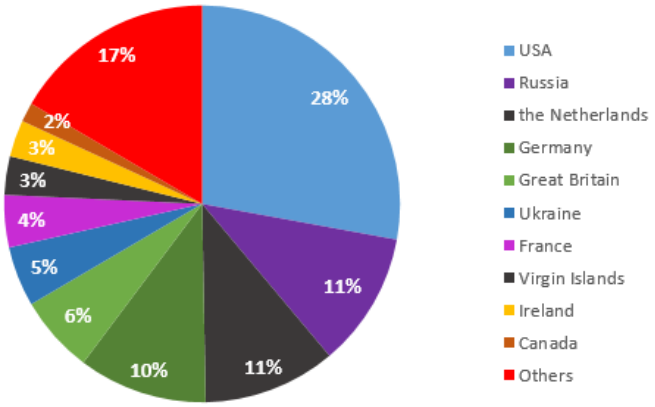


Figure 1. Distribution of main countries hosting malware C&C servers [3]

A. HTTP

Contemporary malware overwhelmingly uses HTTP protocol as a communication channel. It is very pleasant for malware creators because HTTP protocol is enabled on firewalls almost in every organization. The heterogeneity of communication via this protocol facilitates in hiding in common network traffic. A common phenomenon is the use of modified headers of HTTP protocol [17]. Modifications may be expressed by missing field of header, by changed order of header fields and/or by unexpected values of these fields [16].

A typical call from an infected machine to the control server is a HTTP request, usually GET or POST request with specific structure. The call may contain a status code or other information about infected computer (e.g., for example Media Access Control (MAC) address, used character set, etc.). Malware usually attempts to call several different domains while using the same or different URL (see Figure 2).

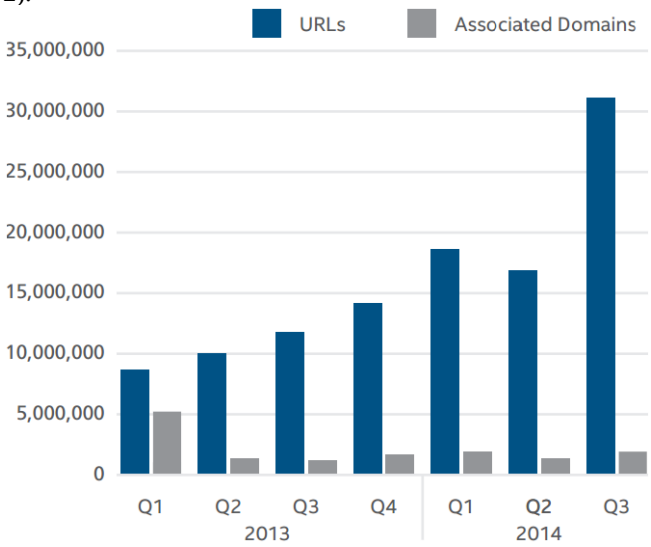


Figure 2. Number of newly discovered suspicious URLs and domains [7]

In this way, it is possible to avoid a blocking at level of domain names and alternatively of IP addresses as well. On the other hand, detection of the same Uniform Resource Identifier (URI) in outgoing calls to reach different domains is very good indicator of anomalies and does not matter if calls are conducted at different times (more often) or in a short period. In the following example, there are HTTP requests of Zeus botnet matched in type, URI and even at 3rd level domain.

- POST hxxp://ix.kasprsky.org/cynic/gate.php
- POST hxxp://ix.dwonkistr.org/cynic/gate.php

Next of possible methods for malware detection is the analysis of values of User-Agent field. This field contains identification data about application that communicates to a server, and it is primarily intended for the server to provide a content to the client applications in various form. In case of malware is often used value which imitates a common browser [17]. In some cases the malware sets this field to its own specific string. It is possible to detect a malware within secured network by evaluating User-Agent values and monitoring where the malware tries to connect. For example, this method may be combined with IP reputation systems. Subsequently, those values of User-Agent that are used for communication with low-reputation IP addresses field are identified as suspicious. In the following example, it can be seen that malware does not try to hide but proudly indicates its presence:

- GET hxxp://www.ody.cc
  HTTP/1.0
  User-Agent: STORMDDOS

 ISBN: 978-1-61208-427-5

In the next example, there is User-Agent field used for a message transfer encoded in Base64 from infected computer to the control server:

- POST hxxp://www.gougle.com
  HTTP/1.1
  User-Agent: UGFzc3dvcmRVc2VybmFtZQ==

B. DNS

It is possible to detect malware actions by collecting and by analyzing DNS requests and their relevant responses. The basic element of detection is pairing domain names and corresponding IP addresses on time. Such information is inestimable for searching the primary cause when investigating of security incident, but it also may help to identify infected computers [8].

Malware tries to contact control servers on IP address obtained from DNS server. A domain name is either permanently written into the malware code or it is generated by pseudo-random algorithm from initial value. Most of control servers exist under several different domain names. A reason for that is an effort to protect a communication channel against common block techniques. If in computer network reveals some computer trying to connect to the control server, it is a good practice to block the targeted IP address and domain to prevent taking out of sensitive data and at the same time prevent receiving next commands from malware operators. However, other infected computers in the computer network may still communicate if are not identified all IP addresses and domain names. Modern fast flux botnets are able to fast change IP address for given domain name for the purpose of increasing robustness. Infected computers contacting control servers in various times are routed to different control servers [10][16].

Database that stores records of DNS translations in time may be very useful. When revealing some domain name, the database can be used to track down IP addresses of several control servers. By reverse procedure, it is possible to identify domains that indicate tracked IP address. For instance, when we put domain *ix.dwonkistrz.org* into the search box on service VirusTotal, we can find that the given domain name corresponds to IP address *195.22.26.231*. We can get a dubious list of domains by searching this IP address at the same website. In addition, we also can find a list of malware which has been spreading in last days.

Malware domains last relatively for a little time. Most of them are registered only a few days before they are used for the first time. Dynamic DNS services are also often used. Domain names are usually created as seemingly legitimate looking (e.g., gmailboxes.com, gougle.eu) or are generated by pseudo-random algorithm from predefined group of keywords (e.g., startftp.com, meown.eu), and/or are generated randomly with limiting conditions (e.g., run30.org, fdms.edu). The first group is primarily intended for confusing users (e.g., may be used in phishing campaign), whereas the other groups prevent simple blocking of domains by security organizations because the percentage of potentially large number of registered domains is very small [11].

By analysis of DNS records, it is possible to detect a range of the incident, but it also enables to reveal incidents which have not been detected yet. It is advisable to investigate those computers which have high failure rates of DNS translations. By this way, the malware may be detected by using generated domain names. An alternative is a calculation of the entropy of a domain name because fully random generated domain names consist of unexpected strings for people [9].

III. OBFUSCATION

A communication between malware and control servers is rarely realized in open form. An encryption may be used, but obfuscation is used more often. The encryption prevents understanding of transferred messages. From the perspective of malware creators, the encryption is limited because of possible recognition of encrypted channel on systems for a detection of range penetration. For example, one of possible techniques for detecting of encrypted data is a calculation of the binary entropy. Encrypted channels to control servers can be found mainly at APT malware. The malware often uses self-signed certificates representing itself as signed certificate by trusted CA [10].

On the contrary, obfuscation typically resorts to simple transformation or a combination of several transformations [9]. The advantage is an easy implementation and resistance against manual inspection. In contrast, performing automatic analysis is relatively simple [12]. It is possible to use brute force and test all commonly performed operations. Afterwards, search expected strings in decrypted data (MAC address, computer name, credentials). Between the most common obfuscations belong:

XOR operations with short key (typically 1-2 bytes)

- To each substring of message of relevant length the XOR function is always applied.

Bitwise or char shift

- A bitwise shift is applied on the message a few positions left or right. Bit writing of message is shifted a few positions left or right. An alternative is a char shifting within a predefined alphabet.

Unique encoding

- A common feature of obfuscation is a conversion of message to Base64 encoding. The encoding is in basic form relatively easily recognizable (e.g., by the end padding). Because of this, in some cases malware creators modify this encoding and change the order of the characters of the coded alphabet.

For example, username *"admin"* has after XOR operation with key *AA* shape *cbcec7c3c4*. After bitwise shift by one bit to the right, it has shape *30b236b4b7* and in Base64 encoding it is *NjE2NDZkNjk2ZQ==* or *YWRTaw4=*. A similar string may occur everywhere in transferred data.

In addition to these classical obfuscation techniques, there are also used targeted modifications distracting an

 ISBN: 978-1-61208-427-5

attention of security analytics. Common is e.g., file transfer with JPG extension, but in fact it is executable file. There are also known cases when malware inserts data into legitimate files. When opening the file an expected content is shown to the user. Everything works and seems to be fine but without detailed analysis it is not possible to reveal these data.

IV. INTERNET SERVICES

There exist many free available web services that can facilitate the analysis of outgoing malware calls. It is not necessary to build an extensive support infrastructure within the organization. Here are several representatives:

VirusTotal (virustotal.com)

- Provides not just a cloud system for analysis of binary files, but also provides services of passive DNS. When searching IP address, it is capable of returning a list of observed domains, which route to the same IP address, and with timestamps of their detection as well. There are also provided information about reputation and trustworthiness. In a similar way works also a query at domain name.

URLQuery (urlquery.net)

- URLQuery (urlquery.net)
  Enables malware control of given URL without a risk of infection of analyst´s computer. It also provides a screenshot of the webpage with an access by a common web browser.

UserAgentString (user-agent-string.info)

- Maintains a list of observed values of HTTP User-Agent field. Enables basic screening of trustworthiness of observed value.

TextMechanic (textmechanic.com)

- Offers a web interface for performing basic transformations with given strings. Enables easy manipulation with the string when there is a suspicion on using obfuscation techniques.

Reputation systems

- A lot of antiviruses and also other security software provide a system for checking up the reputation of the domain name or IP address. When detecting a suspicious calling, a verification of the reputation of a target is a fast indicator if it is necessary to continue the analysis.

Whois

- Provides information about domain registration. From the perspective of malware detection, there are particularly interesting information about recently registered domains or domains with obviously wrong details of responsible person. There are known cases when it was possible to interconnect seemingly unrelated targeted attacks on various organizations on the basis of data in whois.

DISCUSSION

As is shown in Table 1, there are two protocols that are described in Section II. The first protocol is HTTP that, nowadays, is overwhelmingly used by malware as a communication channel in order to merge with standard traffic data because it is a commonly open port in the majority of networks. However, it is possible that a malware to be detected within secure network by evaluating anomalies in HTTP requests, especially by monitoring the User-Agent field and URLs which are used by malware to connect the server. These methods may be combined with IP reputation systems. Subsequently, the values of User-Agent field which are used for communication particularly with understanding the low reputation of IP addresses can be identified as suspicious.

TABLE I. KEY FEATURES OF C&C CHANNELS

Protocol name	Key features
HTTP	• URL monitoring • modified User-Agent field in header
DNS	• pairing domain names and corresponding IP address • database of DNS translations

The second protocol is DNS, which can help to detect symptoms of malware by collecting and analyzing of DNS requests and their relevant responses. The basic element of detection is pairing domain names and corresponding IP addresses on time. By analyzing DNS records, it is possible to detect a range of incidents and also reveal incidents, which have not been detected yet. It is recommended to investigate those computers that have high failure rates of DNS translations. By this way, the malware may be detected by using generated domain names. Another possibility is a calculation of the entropy of domain names because fully random generated domain names consist of unexpected strings for people.

CONCLUSION

The analysis of calls between malware and their C&C servers is an effective method of detection of infected computers. The method can be classified on boundary between anomalous and signature systems. The advantage is revealing of indicators (URI tracks, IP addresses, domain names, values of User-Agent fields, etc.) that subsequently can be searched as common signatures, detecting in this way repeating infections in protected network. The disadvantage is the possibility of false positives. This situation requires at least a basic screening of detected anomalies and subsequent confirmation if it is a manifestation of malware behavior. A seeking of patterns of common malware does not require specialized knowledge. The analysis can be even more facilitated by a range of free available services.

 ISBN: 978-1-61208-427-5

ACKNOWLEDGMENT

This work was supported by grant No. IGA/FAI/2015/037 from Internal Grant Agency of Thomas Bata University in Zlin.

REFERENCES

[1] P. Szor, "The Art of Computer Virus Research and Defense," Addison-Wesley Professional, February 2005.

[2] E. Skoudis and L. Zeltser, "Malware: Fighting Malicious Code," Prentice-Hall, November 2003.

[3] V. Chebyshev, D. Emm, M. Garnaeva, R. Unuchek, D. Makrushin, and A. Ivanov, "IT threat evolution Q3 2014," November 2014. [Online]. Available from: https://securelist.com/analysis/67637/it-threat-evolution-q3-2014/ [retrieved: 7, 2015]

[4] G. O. Gorman, "Google Groups Trojan," 2009. [Online]. Available from: http://www.symantec.com/connect/blogs/google-groups-trojan [retrieved: 7, 2015]

[5] P. James, "Flashback Mac Malware Uses Twitter as Command and Control Center," March 2012. [Online]. Available from: http://www.intego.com/mac-security-blog/flashback-mac-malware-uses-twitter-as-command-and-control-center/ [retrieved: 7, 2015]

[6] D. Talbot, "Dropbox and Similar Services Can Sync Malware," August 2013. [Online]. Available from: http://www.technologyreview.com/news/518506/dropbox-and-similar-services-can-sync-malware/ [retrieved: 7, 2015]

[7] McAfee Labs, "McAfee Threats Report: Q3," November 2014. [Online]. Available from: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2014.pdf [retrieved: 7, 2015]

[8] L. Bilge, E. Kirda, C. Kruegel, M. Balduzzi, and M. Exposure, "Finding Malicious Domains Using Passive DNS Analysis" Network and Distributed System Security Symposium, 2011, pp. 14–42, doi: 10.1145/2584679

[9] Y. He, Z. Zhong, S. Krasser, and Y. Tang, "Mining DNS for Malicious Domain Registrations," Proc. of The 6th International Conference on Collaborative Computing, 2010. [Online]. Available from: http://www.trustedsource.org/download/research_publications/domain_registration.pdf [retrieved: 7, 2015]

[10] Centre for the Protection of National Infrastracture, "Command & Control: Understanding, denying, detecting," 2014. [Online]. Available from: http://www.cpni.gov.uk/documents/publications/2014/2014-04-11-cc_qinetiq_report.pdf [retrieved: 7, 2015]

[11] H. Shulman and M. Waidner, "Towards Forensic Analysis of Attack with DNSSEC," IEEE Security and Privacy Workshops, 2014. [Online]. Available from: http://www.ieee-security.org/TC/SPW2014/papers/5103a069.PDF [retrieved: 7, 2015

[12] S. Shamid, R. N. Horspool, I. Traore, and I. Sogukpinar, "A framework for metamorphic malware analysis and real-time detection," Elsevier: Computers & Security 48, February 2015, pp. 212-233, doi:10.1016/j.cose.2014.10.011

[13] M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing, near-optimal malware specifications from suspicious behaviors," IEEE, Berkeley, CA, USA, April 2010, pp. 45-60.

[14] J. Soryal and T. Saadawi, "Dos attack detection and mitigation utilizing Cross Layer Design," ACM, Ad Hoc Networks, Volume 14, March 2014, pp. 71-83, doi: 10.1016/j.adhoc.2013.11.006

[15] G. Dondossola, F. Garrone, and J. Szanto, "Cyber Risk Assessment of Power Control Systems – A Metrics weighed by Attack Experiments," IEEE, Berkeley, CA, USA, July 2011, pp. 1-9, doi: 10.1109/PES.2011.6039589

[16] K. Aoki, T. Yagi, M. Iwamura, and M. Itoh, "Controlling Malware HTTP Communications in Dynamic Analysis System using Search Engine," IEEE, September 2011, [Cyberspace Safety and Security (CSS), Milan, p. 1-6]

[17] M. Grill and M. Rehák, "Malware Detection Using HTTP User-Agent Discrepancy Identification," IEEE, Atlanta, GA, USA, December 2014, pp. 221-226, doi: 10.1109/WIFS.2014.7084331

 ISBN: 978-1-61208-427-5

Organic Principles to Counter Malware in Automotive Environments

Robert Altschaffel, Sven Kuhlmann, Jana Dittmann, Tobias Hoppe
Arbeitsgruppe Multimedia and Security
Otto-von-Guericke-Universität Magdeburg
Magdeburg, Germany
email: {Robert.Altschaffel|Sven.Kuhlmann|Jana.Dittmann|Tobias.Hoppe}@iti.cs.uni-magdeburg.de

***Abstract*—In an interconnected world malware is not only a topic in classical computer environments (information technology) anymore. Recent examples have shown which damage malware can cause in modern interconnected cyber-physical systems. Based on the increasing threat of malware we propose an approach to counter malware. In contrast to classical approaches like signature scanners or IDS (Intrusion Detection Systems), which focus on detection our approach focuses on the reaction on malware-caused incidents. While these classic approaches usually require manual intervention we focus our approach on an automatic, adaptive reaction. The approach proposed in this work is based on principles of Organic Computing and conceives a self-adapting system for malware reaction in automotive environments.**

Keywords - automotive security; adaptive systems; malware

I. INTRODUCTION

With the ubiquitousness of cyber-physical and interconnected systems a new field of potential targets for malware arises. This work focuses on the automotive domain as an example of an interconnected cyber-physical system. With its broad array of different Electronic Control Units (ECUs) and it's ever increasing range of means to communicate with its environment (Car2X), a modern car is a prime example for both categories. The automotive domain also carries the implication that malfunctions of an ECU could lead to accidents and serious physical harm. Therefore, not only the detection of a malware incident but also a timely, appropriate reaction upon such incidents becomes crucial. While interconnectivity and reliance on electronic components come with the downside of vulnerability, new technology also offers new possibilities to protect the automotive system from such attacks [1].

We propose an adaptive subsystem to support autonomous reactions against malware incidents in the automotive domain. After a brief introduction we give a short overview about anti-malware strategies from Classical computer environments, discuss the special properties of the automotive domain and show general approaches for adaptive systems. In the third section, we introduce our approach for an adaptive malware reaction system while the fourth section shows a first demonstrator of this approach. The fifth section closes with a summary and an outlook.

II. STATE OF THE ART

This section gives an overview about malware detection and reaction in classical computer environment, the characteristics of concurrent and upcoming automotive systems and adaptive systems.

A. Malware in Classical computer environments

In classical computer domains, like Desktop IT malware threats are a central challenge in the productive operation of IT systems. A lot of research went into reducing this threat. Malware analysis (like [2] or [3]) have determined the characteristics of common security threats. Appropriate countermeasures [4] [5] have been established. While protective mechanisms to prevent malware attacks are an important basis of established anti-malware concepts, the vast complexity of today's IT systems with diverse user and/or system interactions make establishing complete protection mere theory. Therefore, an important basis for the treatment of malware incidents is their detection. While classical, signature-based detection strategies are increasingly ineffective due to the ever increasing size of malware samples and bypassing strategies, evolving heuristic approaches still have to cope with false alarms. Consequently, in many cases the subsequent reaction to detected incidents still require manual interaction with human operators – which might be system administrators in professional environments or simply the user himself (e.g. in home environment). Also, in Classical computer environments, the mechanisms for autonomous reactions to malware incidents are a young field of research with little, immature approaches so far.

B. Characteristics of Modern Automotive Systems

Automotive systems differ from classical computer environments in a range of characteristics. In order to design an approach that covers the automotive domain it is necessary to discuss these characteristics and their impact. Important differences originate from the operational environment and the hardware architecture. For the operational environment, we identify the following aspects:

- Safety Implications: In contrast to typical operational environments of classical Classical computer environments, an incident in an automotive system can easily lead to threats to life and health of the user and even passengers and innocent bystanders.

 ISBN: 978-1-61208-427-5

- Technological Aspects: An automotive system is interacting with an analog environment. Therefore many central components within an automotive system have very strict real-time requirements.
- Organisational Aspects: The owner/driver of an automotive system in general is not an IT expert. The automotive system's design therefore must consider that the user is not able to administrate the system or handle user interactions requiring deep, specific knowledge.

These points show that some classic 'emergency reactions' from Classical computer environments - like rebooting or powering down - cannot be applied directly to an automotive environment. Other classical strategies, like software updates during runtime, can only be imported within limits. The hardware architecture itself also has direct impact:

- System Architecture: Like other embedded systems, automotive IT features a broad range of different hard- and software configurations. ECUs in an automotive environment have a much less standardized hardware and software architecture than in a classical computer environment. Also, the usage of processors with small word size and low clock rate is common in automotive environments. Combined with relatively small memory, this seriously restrains available resources. While using this kind of hardware is partially motivated by economic considerations, automotive hardware also needs to cope with various environmental effects virtually unknown in Classical computer environments. Automotive IT needs to be robust against wide variations in temperature, concussions, intruding fluids etc.
- Networking Architecture: The networking technologies and protocols used in automotive IT still differ widely from those utilised in Classical computer environments. In modern cars, the constantly growing number of ECUs require efficient means of interconnection to implement necessary communication use cases. While in the early years of electronic automotive systems, separate analogue lines were drawn for each signal, the complex automotive systems of today require communication via shared media in order to reduce the amount of cables needed (with the aim of reducing weight, costs, difficulties with handling, etc.). Different types of field bus systems are used for this like Controller Area Network (CAN), Local Interconnect Network (LIN), Media Oriented Systems Transport (MOST) and FlexRay. Usage of Ethernet within (parts of) automotive networks is increasingly discussed and tested in the automotive industry. A great difference to Classical computer environments is that typically, automotive networks follow a broadcast strategy, i.e. one bus message of a certain sender is simultaneously received by multiple receivers which evaluate and process its content (if required). Bus systems like CAN (currently the most widely used automotive bus system) are designed for simplicity and efficiency. Unfortunately, such bus systems, which do not provide sender/receiver addresses or even authentication "out of the box", can easily be attacked once access to the internal bus systems has been established. We demonstrated this in previous work [6] [7], as have other research groups like [8] and [9], who base their work on our aforementioned papers.

These characteristics must be taken into account when designing an approach, as discussed in this work.

C. Introduction to Adaptive Systems

In order to deal with as wide a variety of attacks as possible, we need a protective system which is able to adapt to various threats. This system should also be able to adapt to malware which is unknown at the point of the incident. Our approach to design such an adaptive system is based on principles of Organic Computing [10]. Organic Computing is a concept to deal with the ever-growing complexity of hardware, software and networking. While classic engineering approaches try to develop a system by examining all possibilities (like modelling all possible states), Organic Computing goes in a different direction to cope with those cases where the complexity simply grows too vast. Already, developers often do not know which components their system will ultimately interact with – and such cases will increase in future. Instead of preparing different modes for operation in lots of different environments (accepting the risk that the developers did not foresee some of them), a system based on the principles of Organic Computing is able to adapt to different environments by itself –an adaptive system: *„In organic computing, the only task humans hold on to is the setting of goals. As the machine is autonomously organising, detailed communication between programmer and machine is restricted to the fundamental algorithm, which is realising system organisation. Application-oriented mechanisms lose the status of algorithm and are treated as data, in analogy to the transcription factors in the ontogenetic toolbox."* [11]

Obviously, an organic system would need a set of specific properties to behave in such a manner. These properties are referred to as self-x-properties. Various self-x-properties have been proposed, but self-organisation is the core concept characterising adaptive systems. Further examples are self-configuration, self-optimisation or self-healing.

Some of these self-x-properties directly match our goal to deal with new malware trends. Self-healing would best describe our approach to find a fitting reaction to malware incidents. However, the ability for self-perception is the base of any adaptive system – since systems without any information about their state and capabilities could not fulfil any of the self-x-properties. Hence our approach needs to address self-perception, too.

 ISBN: 978-1-61208-427-5

III. ORGANIC PRINCIPLES TO COUNTER MALWARE IN AUTOMOTIVE ENVIROMENTS

In this section we discuss an adaptive system for the reaction against malware incidents in automotive environments.

Automotive environments bring up unique challenges to IT security. They have specific characteristics, and many security incidents carry a broad and deep threat to users and bystanders, including endangering life and limb. This leads to numerous challenges for dealing with malware in automotive environments.

From this perspective, different application scenarios for the implementation of a fitness function arise:

- While interconnectivity is on the rise it is by no means guaranteed that a vehicle will have network access at any time. It could also be possible that a vehicle does not have any access for an extended period of time and so would have no access to necessary updates to block malware.
- Probable safety impacts of an incident make it impossible to simply 'sit it out' or 'ignore it'.

To deal with these challenges we use principles from Organic Computing to design a system which both detects and reacts to malware incidents. Hence, it implements detection and reaction and establishes self-perception and self-healing in an automotive system. As discussed in Section II-B, the individual ECUs used in an automotive environment have serious resource constraints. Therefore, the changes to each of the ECUs need to be relatively simple. In general these ECUs only handle calculations – they get input from some ECUs (or sensors) and give instructions to other ECUs (or actuators). Many of these values are transferred via insecure lines, especially field bus system, throughout the vehicle. An attacker could plug into this system and falsify, or add false values. An input checking on each individual component could detect those tampering attempts. In addition it would distribute the effort to the individual components.

To achieve this we assign a confidence measure (CM) to each source of input value an ECU receives. If the input value seems to be tampered with, the ECU would reduce the confidence in this input source. In order for this to be effective the ECU needs to be able to verify the input given. This is done in two different ways. First, we assign various sources for different ECUs – an ECU that would usually only read speed information will also evaluate the positioning information in order to double check between those two sources. While this approach implements an inter-source sanity check, we also implement an intra-source sanity check, which detects unusual changes in the input data (like the GPS sensor implying that the vehicle moved 100 miles since the last update seconds ago) and reduces the CM accordingly.

The value the ECU works with is calculated from the weighted input values it receives from various sources. If the ECU completely distrusts a certain source it would completely disregard any data from this source.

In addition, the frequency of input messages of a certain type is also monitored. Usually these messages are transmitted in regular intervals - an increase in messages could point out the injection of falsified values. Hence if the number of received messages deviates from the expected frequency, the confidence measure would be decreased. On the other hand the CM of an input source would recover over time if the input is more in line with the other sources again.

IV. IMPLEMENTATION AND EVALUATION OF A FIRST DEMONSTRATOR

This section covers the implementation of a demonstrator to evaluate and refine the approach described in Section III. We introduce the components necessary for our demonstrator, the test setups used, and the evaluation results.

A. Exemplary Simulation Environment

In order to evaluate our chosen approach it was necessary to implement a simulated environment. This environment needs to fulfil a range of various requirements:

- Vehicles consist of a network of different ECUs and actuators. Hence the simulated vehicle needs to consist of multiple ECUs and means for them to communicate with each other.
- The simulated vehicle needs to be modular so more components can be added in order to increase the complexity and feasibility of the simulation. This also allows for different test cases with alterations to the adaptive subsystems, different attack vectors or malfunctions.
- The simulation environment must be able to support the evaluation by offering exhaustive output to ensure traceability of the simulation's results.

Our implementation follows a modular approach with different types of modules as shown in Figure 1. The main component is the simulator itself which handles the physics inside the simulation and contains the 'real' system state of the vehicle. Linked to the simulator is the logging subsystem. This subsystem records the information needed for the evaluation.

Our modelled vehicles themselves consist of actuators, sensors, assistance systems and a bus. Sensors get information about the current state from the simulator. Actuators on the other hand give feedback about the actions of the vehicle to the simulator. Assistance systems are the various ECUs which use sensor information to control actuators. The communication between sensors, actuators and ECUs is covered by a communication bus which mirrors a field bus used in automotive networks.

For our first exemplary simulation environment we implemented the simulator itself, the bus, various actuators, sensors and assistance systems. As assistance systems we implemented cruise control (CC), adaptive cruise control (ACC), lane-keeping assist, anti-lock braking systems (ABS) and park distance control (PDC).

 ISBN: 978-1-61208-427-5

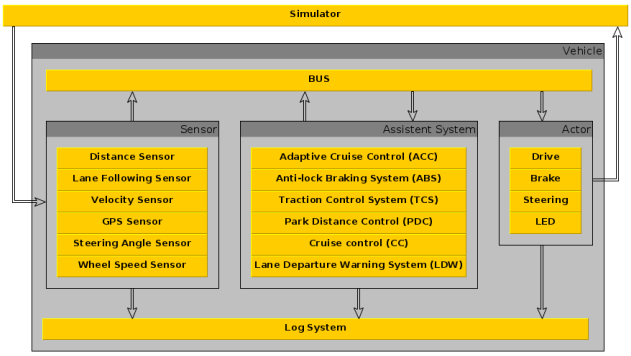


Figure 1. Architecture of our Demonstrator

B. *Exemplary Scenarios and Attacks*

The modular approach for our simulation environments allows for the easy simulation of various attacks or malfunctions within the simulated car (or cars). In order to evaluate the usefulness of our approach we formulated various scenarios:

- S1 - Defunct sensor: In this scenario the GPS sensor is manipulated in a way that it is not able to supply data anymore. This might be due to a malware attack, removal of the component or simple breakdown of the component.
- S2 - Malicious messages: In this scenario an additional component is added which adds false input values to the bus. In our case we choose the GPS sensor as main target and hence inject falsified GPS data. These injections occur at the same frequency the correct GPS data is transferred.
- S3 - Flooding attack: In this scenario we perform a basic Denial-of-Service (DoS) attack on the assistance systems, which rely on the input from the GPS sensor. We add a malicious component, which floods the bus with random GPS input data.

C. *Evaluation results*

After implanting and testing the various scenarios we could evaluate them using the data supplied by the logging subsystem.

- S1 - Defunct sensor: Over the course of time the CM of the GPS sensor went down to zero and other sources (velocity, steering angle) were used by the assistance systems relying on position data.
- S2 - Malicious messages: The assistance systems start to notice that the input given by the GPS sensor is not plausible and therefore reduced the corresponding CM. Instead they relied more on the other sources (velocity sensor, steering angle) as input. When the CM of the GPS sensor went to zero it was ignored completely.
- S3 - Flooding attack: The strong divergence from the amount of expected messages led to a quick decay of the CM of the GPS sensor. After mere moments the sensor data was not used anymore. After the flooding stopped the CM slowly recovered.

V. Conclusion and Outlook

In this work we propose a novel approach to include an adaptive reaction system against malware incidents in automotive systems. Most current approaches as established in Classical computer environments only handle the detection of a malware incident and require a manual reaction. Furthermore they rely only on pre-defined knowledge from external sources for detection strategies. With the approach presented here, an automotive system is able to adapt to new malware threats by itself. This approach not only handles malicious manipulations but also unintentional malfunction like sensor failure, for example.

However, at this point this approach is very juvenile, and several points need to be addressed in the future. For example, further potential use cases will require more complex heuristics to achieve a useful determination of the CM of different input sources. Also, a more complete evaluation will be required to get a broader impression of the gains of the proposed approach. We consider doing so in future research, either by further extensions to the presented simulation environment or by implementing the methods proposed in this work in a real automotive (laboratory) system.

Acknowledgements

We like to thank our students from our course "Praktikum IT-Sicherheit 2014/2015" for their work concerning the presented demonstrator.

This work was partly supported by German Research Foundation, project ORCHideas (DFG GZ: 863/4-1).

This work was also partly supported (definition of the scenarios derived from high level project requirements) by European Research Foundation, project SAVELEC (FP7 - SEC-2011, Grant Agreement Number 285202).

References

[1] J. Dittmann, T. Hoppe and C. Vielhauer, "Multimedia Systems as Immune System to Improve Automotive Security?", SAFECOMP 2013, Toulouse, France, 2013.

[2] James M. Aquilina, Eoghan Casey and Cameron H. Malin, "Malware Forensics: Investigating and Analyzing Malicious Code", Elsevier, ISBN 987-1-59749-268-3, 2008.

[3] M. Sikorski and A. Honig, "Practical Malware Analysis – The Hands-On Guide to Dissecting Malicious Software", No Starch Press, San Francisco, ISBN 978-1-59327-290-6, 2013.

[4] E. Skoudis and L. Zeltser, "Malware – Fighting Malicious Code", Prentice Hall International, ISBN 978-0131014053, 2003.

[5] P. Mell, K. Kent and J. Nusbaum, "Guide to Malware Incident Prevention and Handling", National Institute of Standards and Technology Special Publication 800-83, November 2005. http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf (last access: 14/01/2015), 2005.

[6] Tobias Hoppe, Stefan Kiltz and Jana Dittmann, "Security threats to automotive CAN networks – practical examples and selected short-term countermeasures", Computer Safety, Reliability, and Security, Proceedings of the 27th International Conference SAFECOMP 2008, Newcastle, UK, September 2008; Springer LNCS 5219; S. 235-248; Editors:

 ISBN: 978-1-61208-427-5

Michael D. Harrison, Mark-Alexander Sujan; ISBN 978-3-540-87697-7, 2008.

[7] Tobias Hoppe, Stefan Kiltz and Jana Dittmann, "Automotive IT-Security as a Challenge: Basic Attacks from the Black Box Perspective on the Example of Privacy Threats", Computer Safety, Reliability, and Security, Proceedings of the 28th International Conference SAFECOMP 2009, Hamburg, Germany, September 2009; Springer LNCS 5775; S. 145-158; Editors: Bettina Buth, Gerd Rabe, Till Seyfarth; ISBN 978-3-642-04467-0, 2009.

[8] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, et al.: "Experimental Security Analysis of a Modern Automobile", The IEEE Symposium on Security and Privacy, Oakland, CA, May 16-19, 2010.

[9] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, et al.: "Comprehensive Experimental Analyses of Automotive Attack Surfaces", In D. Wagner, ed., Proceedings of USENIX Security 2011. USENIX, Aug. 2011.

[10] C. Müller-Schloer, H. Schmeck, T.Ungerer, "Organic Computing — A Paradigm Shift for Complex Systems, Springer" , ISBN: 978-3-0348-0129-4, 2011.

[11] R.P. Würtz (ed.), "Organic Computing. Understanding Complex Systems", doi: 10.1007/978-3-540-77657-4 1, © Springer-Verlag Berlin Heidelberg 2008

 ISBN: 978-1-61208-427-5

Apate - A Linux Kernel Module for High Interaction Honeypots

Christoph Pohl
MuSe - Munich IT Security Research Group
Munich University of Applied Sciences
Munich, Germany
Email: christoph.pohl0@hm.edu

Michael Meier
Fraunhofer FKIE Cyber Defense
University of Bonn
Bonn, Germany
Email: michael.meier@fkie.fraunhofer.de

Hans-Joachim Hof
MuSe - Munich IT Security Research Group
Munich University of Applied Sciences
Munich, Germany
Email: hof@hm.edu

***Abstract*—Honeypots are used in IT Security to detect and gather information about ongoing intrusions, e.g., by documenting the approach of an attacker. Honeypots do so by presenting an interactive system that seems just like a valid application to an attacker. One of the main design goals of honeypots is to stay unnoticed by attackers as long as possible. The longer the intruder interacts with the honeypot, the more valuable information about the attack can be collected. Of course, another main goal of honeypots is to not open new vulnerabilities that attackers can exploit. Thus, it is necessary to harden the honeypot and the surrounding environment. This paper presents Apate, a Linux Kernel Module (LKM) that is able to log, block and manipulate system calls based on preconfigurable conditions like Process ID (PID), User Id (UID), and many more. Apate can be used to build and harden High Interaction Honeypots. Apate can be configured using an integrated high level language. Thus, Apate is an important and easy to use building block for upcoming High Interaction Honeypots.**

Keywords–Honeypot; Intrusion Detection; Linux Kernel; Rule Engine

I. INTRODUCTION

Honeypots are well known tools for Intrusion Detection and IT Security research. Usually, honeypots fall into one of two classes: Low Interaction Honeypots and High Interaction Honeypots. A Low Interaction Honeypot simulates attackable services, systems, or environments whereas a High Interaction Honeypot [1][2] offers a real exploitable service, system, or environment. As in most cases a honeypot is not a productive system, every activity on a honeypot is either unintended use or an attack.

When deploying a High Interaction Honeypot, it is necessary to harden the honeypot to avoid attackers gaining unintended control of the system running the honeypot. A High Interaction Honeypot should by definition be exploitable, but it should prevent annoying or harmful operations on the honeypot system. Another important requirement for High Interaction Honeypot is to log as much information as possible about the state of the system and about ongoing intrusions. Therefore, a High Interaction Honeypot needs a highly flexible way to decide which information should be logged and which should not. Apate offers such a flexible way, using a high-level language for configuration. Also, it should be possible to log information on a as fine granular level as possible. Apate offers a logging on system call level. Manipulation of system calls, depending on user interaction or the system environment, is necessary to provide High Interaction Honeypot functionalities. This allows the honeypot provider to present different environments depending on PID, UID (and many more), or system call parameters. For example, the High Interaction Honeypot provider is able to present one file structure to PID 42 and a completely different file structure to PID 43. This manipulation can be used to decoy an attacker. Furthermore, it can also be used to suppress harmful actions. The honeypot admin is able to prevent execution of system call. Blocking a system call can be done by really blocking (not calling the real system call), or in a more sophisticated way. At last, it is necessary that High Interaction Honeypot components (like the proposed LKM) should be hard to detect for intruders. This requirement calls for sophisticated technologies, already known from rootkits. For productive use, it is necessary that a High Interaction Honeypot module has only low computational overhead. An attacker should not be able to detect a High Interaction Honeypot by observing performance leaks.

Apate is a Linux Kernel module that fulfills all requirements mentioned above. Hence, it is an important building block for High Interaction Honeypots.

The rest of this paper is structured as follows: Section II provides an overview on related work. Section III describes the design and implementation of Apate in detail. Section IV shows the evaluation of Apate. Section V concludes the paper.

II. RELATED WORK

A well known honeypot tool, based on LKM for 2.6 Linux Kernel, is Sebek [3][4]. Sebek is primarly used for logging purposes in High Interaction Honeypot. Thus, it provides several methods for detailed logging (like logging via network or GUI). In [5][6], ways to detect Sebek are described. Sebek does not provide the possibility to manipulate system calls, hence it does not offer such a fine-grain information logging as provided by Apate.

Another approach for monitoring systems is to use virtual machine introspection and system view reconstruction. For example, [7], [8], and [9] use this approach. Introspection realized on hardware level of the virtual machines offers a stealthier approach then Apate. However, Apate provides additional means to manipulate the behavior of system calls, which are not supported by [7], [8], and [9], hence Apate is superior to these approaches.

SELinux [10] is a well known tool for inserting hooks at different locations inside the kernel. Such an approach provides

 ISBN: 978-1-61208-427-5

access control for critical kernel routines. SELinux can be controlled on a very fine granular level with an embedded configuration language. Although SELinux is very useful in hardening a kernel, it is not designed for honeypot purposes. Especially, it lacks in the possibility to decoy the attacker using "wrong" information.

Grsecurity [11] with PAX [12] is similar to Apate. However, it greatly differs in ease of deployment and ease of configuration [13]. It also lacks in the possibility to decoy the attacker with "wrong" informations.

In conclusion, non of the mentioned related work fulfills all requirements listed in Section I. Apate fulfills all requirements, hence is a useful building block for upcoming High Interaction Honeypots.

III. Design and implementation

Apate intercepts system calls and allows to execute custom code in these calls. Figure 1 shows the interception strategy of Apate.

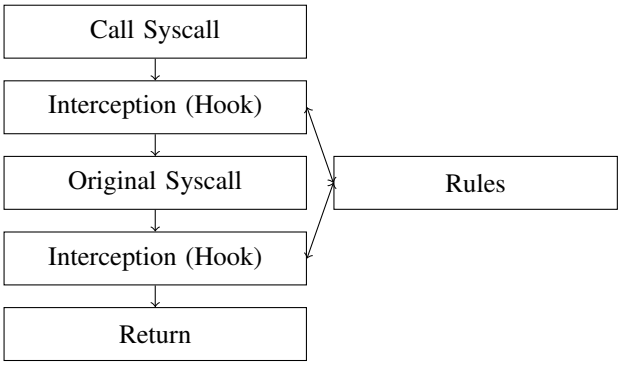


Figure 1. interception strategy of Apate

Apate does not manipulate the syscall table to prevent detection (see Subsection III-D for details). Apate intercepts the syscall within the syscall target, i.e., the real syscall address is called but Apate jumps immediately to the interception routine (after consuming some decoy assembler code). The hook decides on the action to invoke, based on the rules for this system call. Within this action, it is able to manipulate, block and/or log a system call. The following hooks are implemented in Apate :

- sys_open, sys_close, *sys_open*
- sys_read, sys_write, sys_unlink
- sys_execve
- sys_getpid, sys_getuid
- sys_mkdir, sys_rmdir
- sys_getdents

This paper focuses on the usage of system calls that are related to File IO and execution control as these system calls are usefull for hardening High Interaction Honeypots.

A. Configuration

Apate can be configured in a very flexible way as can be seen in Figure 2. The configuration file *rules.apate*, written in a high level language (see sectionIII-B for details), gets compiled by the Apate compiler, resulting in the file *apaterules.c*. Together with the original source code, the compiler generates the Apate LKM. The resulting LKM can be loaded into kernel

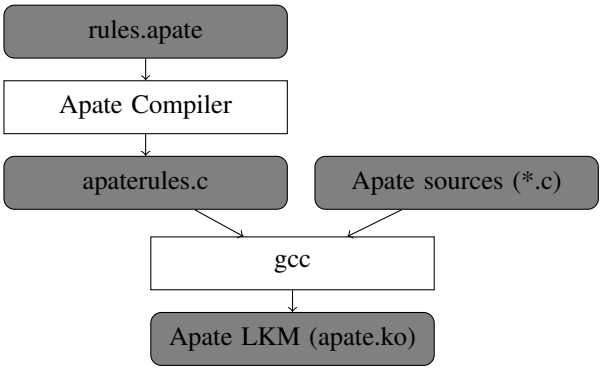


Figure 2. Configuration workflow of Apate

with common insmod util. Once loaded, the ruleset is active.

The configuration consists of rulesets. A ruleset is an ordered list of rules. A system call gets intercepted when one or more rules match. One system call can have more than one matching rule with different decision parameters. There are three major types of decision parameters:

- Parameters that are system call independent like PID, UID, SSID (in fact every variable from *struct task_struct* [14] could be used for conditions).
- Parameters that are dependent to the specified system call. Often, these are function parameters like paths.
- Parameters that are defined by functions. This decision parameter allows to build reactive systems. For example, one can define a condition which reads some file, whenever the file contains a keyword the condition could be true.

Rules are defined as stated below:

Let be $true = 1$ and $false = 0$. Let $c(a, b)$ be a condition such that:

$$\begin{aligned} C &: A \times B \to \{0, 1\} \\ (a, b) &\mapsto c(a, b) \end{aligned} \tag{1}$$

, where $a \in A$ and $b \in B$ are two parameters, which are used by $c()$ for the calculation of the condition. The parameters are further called decision parameters. For example, a decision parameter could be the path of the system call sys_open*()* and the related condition is *if (param[0] == "/etc/passwd") ? 1 : 0* where $a = param[0]$ and $b = "/etc/passwd"$. This rule matches system calls trying to get access to the file $"/etc/passwd"$

Let $cb(d, e, f)$ be a condition block. A condition block calculates the result of conditions or other conditionblocks with *AND* or *OR*. A condition block uses two parameters d and e and an operator f. d and e can be the result of any $c(a, b)$ or another $cb(d, e, f)$.

CB is the set of all possible condition blocks:

$$\begin{aligned} CB &: \{0, 1\} \times \{0, 1\} \times \{AND, OR\} \to \{0, 1\} \\ (d, e, f) &\mapsto cb(d, e, f) \end{aligned} \tag{2}$$

 ISBN: 978-1-61208-427-5

Further, the set of all conditions and condition blocks is AC such that

$$AC = C \cup CB \tag{3}$$

For $cb(d, e, f)$ let $d, e \in AC$ and $f \in \{2, 4\}$. When $f = 2$ the operator AND will be used. When $f = 4$ the operator OR will be used. $c^*(a, b)$ is a condition that returns always true. The second parameter in the conditionblock can be neutralised with $cb(d, c^*(0, 0), 2)$

This leads to the definition

$$cb(d, e, f) = \begin{cases} 1 & \text{if } (d + e) * f \geq 4 \\ 0 & \text{other} \end{cases} \tag{4}$$

This definition makes it possible to group different conditions and to be aware of precedences.

Let A be the set of atomic actions. An atomic action is a function that provides only one single functionality. For example, an atomic action can be the redirection of a system call. An action $a \in A$ falls in one of three groups:

- Manipulating actions
- Logging actions
- Blocking or emergency exit actions

Let AS be an orderd list of actions. The index function $i(x)$ assigns an index to each element $x \in AS$, hence

$$AS = \{x \in AS | 0 < i(x - 1) < i(x)\} \tag{5}$$

Let AAS be the lists of all actions. A rule $r^{g,h}$ consists of one condition block $g \in CB$ and an action set $h \in ASS$. Let R be the set of all rules. Whenever the condition block returns 1, the action set h is started.

Let RS be a list of sorted rules ($RS \in R$). Each element of RS has a flag fl. A flag is defined as

$$fl \in \{exit = 1, \neg exit = 0\} \tag{6}$$

When a system call gets called, all rules in RS are calculated beginning with the first rule in RS and until a rule is in state $true$ and $fl = 1$.

Using the definitions above, a highly configurable system could be build. Including some basic predefined conditions enhances convenience, e.g., equality checks for integer, floats or strings.

B. Configuration High Level Language

The configuration language implements two main requirements: first, the configuration language should be flexible, including the ability to reuse patterns, store variables, calculate with operators, embed external functions, define functions, and use decision statements. This allows to use the language to describe even very complex scenarios. Second, the configuration language should provide a transparent way to define rules, related to honeypots (or in scope of this paper to control and manipulate system calls). To deal with these requirements, the Apate language combines concepts known from functional programming (in this case Haskell [15]) with a concept well known from packet filter configuration (in this case pf [16]).

This Section gives a brief introduction to the important parts of the language. For the sake of clarity, some convenience features of the Apate language (e.g., embedded C, self defined functions, loops) are omitted.

Listing 3 shows some example source code for the Apate language.

```
define c1,c2,c3 as condition
define r1,r2 as rule
define a1,a2 as action
define cb1 as conditionblock
define rc1 as rulechain
define sy1 as syscall

let c1 be testforpname
let c2 be testforparam
let c3 be testforuid
let a1 be manipulateparam
let a2 be log
let sy1 be sys_open

let cb1 be {(c1("mysql") && c2(0;"/var/\
    lib/mysql/*"))}

let r1 be {cb1->a1(0;"/var/lib/mysql/*" \
  ;"/honey/mysql/")}
let r2 be {{c3(">",0)}->a2()}
let rc1 be {r2,:r1} // :defines exit

bind rc1 to sy1
```

Figure 3. Example Sourcecode Apate language

The first block with the `define` statements binds variables to different types (like condition, rules, or functions). The code block with the `let` statements points these variables to values or functions. In this case, it defines 3 conditions (`c1,c2,c3`). `c1` will test the actual process name against another string. `c2` tests if a param of the actual syscall is equal to a given value. `c3` tests if the actual uid is equal to a given value. `a1,a2` are actions. `a1` manipulates a parameter of the actual system call. `a2` logs a system call. The variable `cb1` represents a condition block. Its `let` assignment also shows that it is possible to write nested variable assignements. In this case, the conditions `c1,c2` are combined with `&&` (*AND*). In the same line, the conditions `c1,c2` gets assigned with parameters. In this case the condition `c1` checks if the current parent process is the mysql-Process. `c2` checks if the first parameter (`0`) of the current system call is equal to */var/lib/mysql/** . The asterisk describes a wildcard function. The rule assignment for `let r1 be...` binds a conditionblock to an action. In this case, it means whenever the conditionblock returns true the action `a1` rewrites the first param of the current system call. It replaces */var/lib/mysql* with */honey/mysql* . The rule `r2` logs the current system call whenever the current UID is greater than 0. A ruleset (rulechain) `rc1` is assigned with `r2,r1`. The `r1` rule is also assigned as exit rule (`..:r1..`). When this rule fires, no further rules will be called. In the last line, the rule chain `rc1` is bound to the system call *sys_open*.

In conclusion, when the system call *sys_open* gets called,

the parent process is the mysql process and the system call parameter (in this case the path which should be opened) begins with */var/lib/mysql/**, this syscall gets manipulated and the syscall will open a file under */honey/mysql/...* . The second rule means that every call for *sys_open* will be logged, except when the root user calls this system call.

C. Manipulation of System Calls

If a rule matches, the corresponding action chain gets called to manipulate the original system call. An action chain has a length l with $1 \leq l < n$.

Figure 4 shows an example for the manipulation strategy. Functions prefixed with *f_* are actions.

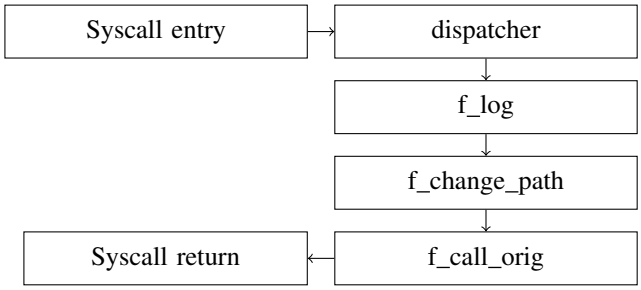


Figure 4. Conceptual Manipulation Strategy

The dispatcher represents the rule engine, deciding which action chain should be used. In this example the first action logs the system call. The next action manipulates some parameter like a path or anything else. The *f_call_orig* calls the original system call with the manipulated parameter. The result gets returned to the callee.

Technically, one action is a function that consumes all system call parameters, including the current *struct task_struct* and a pointer to the *syscall_result* variable. Each function returns an Integer, indicating wether the function call has been successful or not. Whenever a function returns an error, the action chain gets disrupted and an error routine is called. Finally, the hook returns the *syscall_result*. In case of an error the system call returns a system call dependent error.

D. Hiding Hooks and LKM

An attacker should not be able to detect Apate, otherwise Apate would not be suitable for High Interaction Honeypots. As hiding software in all use cases is very difficult, Apate must at least hide itself until the effort of detection of Apate is unreasonable high for an intruder. This requires to define which effort is unreasonable high for an attacker and which is not. The following actions are defined as reasonable for an attacker, hence should be prevented:

- Testing for module presence with standard utils like *lsmod,modinfo,...* or misleading errors when using *insmod* and similar tools
- Testing for presence of module in */proc/module* and */sys/module*
- Testing for presence of Apate related logfiles, configurations, and other artifacts

To hide Apate, it is necessary to remove the module from the module list. Simplified, all modules are represented in a global linked list. By using

```
list_del_init(&__this_module.list);
```

the module is removed and therefore invisible. To hide from the */sys/module* Apate uses

```
kobject_del(&THIS_MODULE->mkobj.kobj);
```

to remove itself from this representation. With these modification, the module is invisible to standard utils (they use */proc/module*) and in */sys/module* . These technologies are also well known rootkit technologies see for example [18][19].

Apate does not use any configuration files beyond the configured rules. The high level language should be deleted by the honeypot admin after its compilation into Apate. Hence, Apate cannot be identified by an attacker looking for configuration files.

Apate is used to cloak logfiles: predefined rules in Apate prevent all users to see, read, or write Apate logfiles. To gain access to the logfile, a system administrator need to restart the host system without the honeypot.

To detect a hook, an intruder needs to analyze physical memory. Apate makes it hard to load a new module into the kernel. It prevents to load another kernel module by overriding the flag that controls the module loading ability. Beyond the possibilities of Apate, the honeypot admin can harden the host system to ensure that this dumping has a high effort for an intruder.

Apate has different opportunities to insert hooks into system calls. By default, Apate changes the function pointer in the system call table. This is sufficient as long as the intruder has no possibility to compare the original table with the hooked table. If this is not enough protection, the admin can decide to harden the system with some anti-rootkit technologies. This makes it impossible for Apate to overwrite the jump points. Figure 5 shows the alternative hooking technology. This

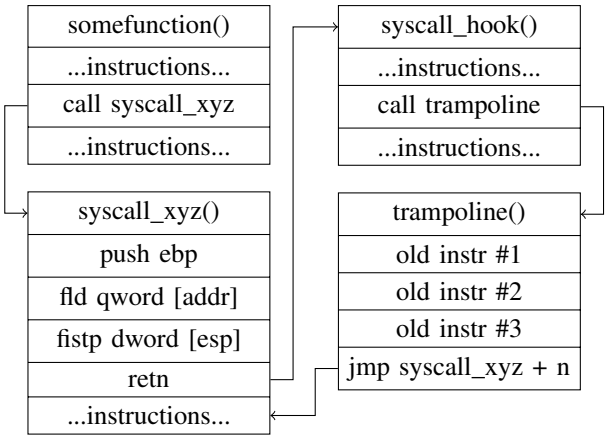


Figure 5. Hooking using a so-called trampoline

technology is well known from Windows and Linux rootkits. During the hooking process, Apate stores the first n bytes of the target system call function. The stored commands will be copied to a trampoline function. Instead of the original commands, Apate injects a jump operation. This lets the process jump into the hooking function immediately after entering the original system call function. Whenever the hooking function calls the original system call, it calls the trampoline function.

 ISBN: 978-1-61208-427-5

The original code is executed, then the trampoline function lets the process jump into the original function with an offset of n bytes. The trampoline is a feature to obfuscate the hook for rootkit detection tools and uses live patching technologies. Thus, it can be detected with core dump disassembling. This is out of scope of this paper as it is assumed that the effort to detect the honeypot with disassembling tools is too high.

IV. EVALUATION

There are three major goals for Apate. The first goal is to provide a highly flexible configuration. Although the proposed configuration system works well and is suitable for High Interaction Honeypots, it must be ensured that every possible combination of rulesets and actions can be described. This means that the configuration language must be turing complete. For this, it is determined that an action a is able to decide which rule from the ruleset should be invoked next. This means it can jump to any other rule from a given ruleset. It is also determined that Apate has an array (in this case an impossible array with infinite indices which can hold any other type (like actions, rules, other defined variables or anything else)). Technically, Apate has a register and a stack. Last, it is determined that an action is able to fill or read any index of this array. Together with actions for calculations and conditions for jump decisions, the system is turing complete. In fact any action is just a C-Function and the *define* statement creates variables.

The second goal is to provide a system which achieves a suitable level of stealthiness. As described in Subsection III-D, the system hides itself from common util tools like *lsmod, modinfo, modprobe, insmod* . Apate is also not available in */proc/module* or */sys/module* . To test for presence in any logfile a simple grep command with typical signatures for Apate (Simplified each log entry or configuration includes the string "apate", thus it is easy to detect it) is fired on the full system. However with proper rules these log entries are not visible by standard system commands.

The third goal is that Apate should be efficient. Performance tests should assure that Apate is able to serve under productive usage scenarios. The most important performance factor is the overhead of logging. To evaluate the performance of Apate in a productive scenario, the execution time of *sys_open* , *sys_write* ,*sys_read* , and *sys_close* are measured. The *sys_open* and *sys_close* get called just once a file is opened or closed. The *sys_write* and *sys_read* get called more often (under the condition that heavy writing will be done on the system). Thus, the test pattern concentrates on *sys_write* and *sys_read* . For the performance evaluation, data is copied from one file to another using increasing file lengths. This will be done for 100 times for each file size. The source file is generated on the fly from */dev/random* before each copy command. After each successful copy command the target file is deleted. A Gentoo 64 Bit system with 32 GB Ram and 16 Cores is used for all performance tests. The kernel is optimized by disabling unnecessary drivers and by enabling some debugging flags. One source file is generated for each size with random bits and a length of $l(file)$ bytes. Let the size of the file be:

$$0 < l < 1,000,000,000 \tag{7}$$

TABLE I. PERFORMANCE MEASUREMENT

Measurement	m_1	m_2	m_3	m_4
Measurements	110,800	110,800	110,800	110,800
Unique Filesizes	1,108	1,108	1,108	1,108
sd(runtime sec)	0.1066	0.2421		0.2452
var(runtime sec)	0.0114	0.0586		0.0601
iqr(runtime sec)	0.0010	0.0026		0.0023

and

$$l_n(file) = \begin{cases} l_{n-1} + 1 & \text{if } l_n < 1,000,000 \\ l_{n-1} + 1,000 & \text{if } 1,000,000 \leq l_n \\ & \wedge l_n < 100,000,000 \\ l_{n-2} + 1,000,000 & \text{if } 100,000,000 \leq l_n \\ & \wedge l_n \leq 1,000,000,000 \end{cases} \tag{8}$$

Four different settings were tested.

The first setting (m_1) is used as reference setting. It does not use any interception.

The second setting, m_2, uses only one rule which always returns true. The related action set calls the origin system call and logs this action. This is the shortest way in Apate to provide logging functionality. This testing is used to evaluate the logging overhead of Apate.

The third and fourth setting, m_3 and m_4, evaluate the influence of rules. Each rule consists of 50 conditions with $\{c_0, c_1, \ldots, c_{50}\}$ where each condition is combined with an *and* statement. The last condition returns false. Each test uses 50 rules. The last condition in rule number 50 (last rule) returns true. Overall, each system call passes 2500 conditions. This triggers an action set that will call the original system call (m_3 and m_4) and then logs this action (only m_4).

Table I shows the results of the performance evaluation. The *sd* -row shows the standard deviation, *var* shows the variance, and *Iqr* shows the interquartile range.

Figure 6 shows the correlation between file size and runtime. For every curve the median of the measured runtimes for each unique file size is connected with a line. The m_1 curve shows the reference setting. The m_2 curve shows that the logging component has a big influence on performance. Each syscall and its values were logged. Each log was sent to another server using UDP. Gentoo uses a buffer with 65,365 Byte. To copy a file with one Gigabyte it needs 32,720 syscalls. This explains the overhead of m_2 and m_4. The m_3 curve shows that the rule engine works with just a small overhead when only conditions get processed. For one measurement with a file size of one Gigabyte, the engine processed 81,800,000 conditions. However, to copy a file with less than 65,365 Byte only 4 syscalls are passed and therefore only 10,000 conditions gets processed.

In conclusion, these measurements shows that it is possible to build a syscall interception framework which is able to provide proper configuration with acceptable overhead. The evaluation does not show a single case that prevents a productive usage of Apate.

 ISBN: 978-1-61208-427-5

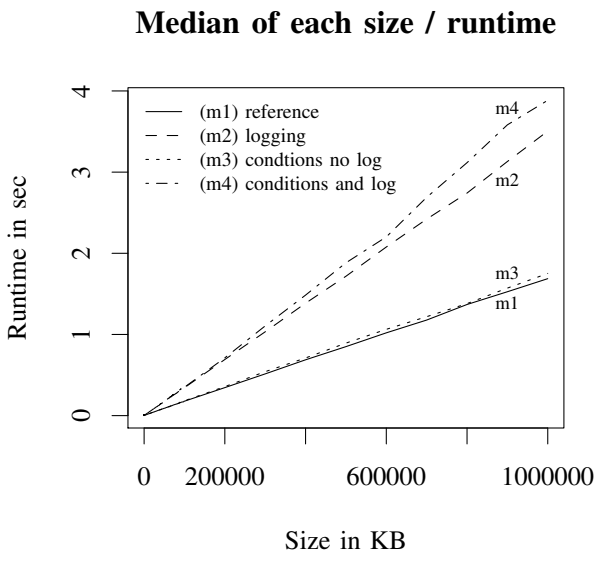


Figure 6. Relation between runtime/filesize/rules *sys_open*

V. CONCLUSION

This paper presented Apate, a Linux Kernel Module for hardening High Interaction Honeypots. Apate works on a system call level, is able to log, block and manipulate these calls, and uses an easy to use yet powerful configuration language. The evaluation shows that Apate has a moderate performance overhead and can be used in productive honeypot systems. Apate is also stealthy enough for most common usage scenarios. Overall, Apate is an ideal basis and important building block for upcoming High Interaction Honeypot Systems.

REFERENCES

[1] C. Pohl and H.-J. Hof, "The All-Seeing Eye: A Massive-Multi-Sensor Zero-Configuration Intrusion Detection System for Web Applications," in SECURWARE 2013, The Seventh International Conference on Emerging Security Information, Systems and Technologies, 2013, pp. 66–71.

[2] C. Pohl, A. Zugenmaier, M. Meier, and H.-J. Hof, "B.Hive: A Zero Configuration Forms Honeypot for Productive Web Applications," in International Conference on ICT Systems Security and Privacy Protection (IFIP SEC 2015), 2015.

[3] Honeynet Project, "Know Your Enemy: Sebek," 2003. [Online]. Available: http://old.honeynet.org/papers/sebek.pdf

[4] E. Balas, "Sebek: Covert Glass-Box Host Analysis," ;login: THE USENIX MAGAZINE, no. December 2003, Volume 28, Number 6, 2003. [Online]. Available: https://www.usenix.org/publications/login/december-2003-volume-28-number-6/sebek-covert-glass-box-host-analysis

[5] T. Holz and F. Raynal, "Detecting honeypots and other suspicious environments," in Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005. IEEE, 2005, pp. 29–36. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1495930

[6] M. Dornseif, T. Holz, and C. Klein, "NoSEBrEaK - Attacking Honeynets," in Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, Jun. 2004. [Online]. Available: http://arxiv.org/abs/cs/0406052

[7] C. Song, B. Ha, and J. Zhuge, "Know Your Tools: Qebek – Conceal the Monitoring — The Honeynet Project," http://www.honeynet.org/papers/KYT_qebek, visited 25.02.2015. [Online]. Available: http://www.honeynet.org/papers/KYT_qebek

[8] T. K. Lengyel, J. Neumann, S. Maresca, B. D. Payne, and A. Kiayias, "Virtual machine introspection in a hybrid honeypot architecture," in CSET'12: Proceedings of the 5th USENIX conference on Cyber Security Experimentation and Test. USENIX Association, Aug. 2012.

[9] X. Jiang and X. Wang, ""Out-of-the-Box" Monitoring of VM-Based High-Interaction Honeypots," in Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2007, pp. 198–218. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-74320-0_11

[10] S. Smalley, C. Vance, and W. Salamon, "Implementing selinux as a linux security module," NAI Labs Report, vol. 1, no. 43, 2001, p. 139.

[11] Open Source Security, "grsecurity," https://grsecurity.net, 2015, visited 25.02.2015.

[12] PAX Team, "Pax," https://pax.grsecurity.net, 2015, visited 25.02.2015.

[13] M. Fox, J. Giordano, L. Stotler, and A. Thomas, "Selinux and grsecurity: A case study comparing linux security kernel enhancements," 2009.

[14] L. Torvalds, "Linux kernel release 3.x source linux/sched.h," https://github.com/torvalds/linux/blob/master/include/linux/sched.h, 2015, visited 25.02.2015.

[15] S. Marlow, "Haskell 2010 language report," https://www.haskell.org/onlinereport/haskell2010/, 2010, visited 25.02.2015.

[16] OpenBSD, "Pf: The openbsd packet filter," http://www.openbsd.org/faq/pf/, 2015, visited 25.02.2015.

[17] C. Pohl, "Github apate sourcecode gpl2," https://github.com/c00clupea/apate, 2015, visited 25.02.2015.

[18] M. H. Ligh, A. Case, J. Levy, and A. Walters, The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. John Wiley & Sons, 2014.

[19] T. B. (TurboBorland), "Modern linux rootkits 101," http://turbochaos.blogspot.de/2013/09/linux-rootkits-101-1-of-3.html, 2013, visited 25.02.2015.

 ISBN: 978-1-61208-427-5

Automatic Human Tracking using Localization of Neighbor Node Calculation

Tappei Yotsumoto[†‡], Kozo Tanigawa[†], Miki Tsuji[‡],
Kenichi Takahashi[‡], Takao Kawamura[‡], Kazunori Sugahara[‡]

[†]System Engineering Department,
Melco Power Systems Co. Ltd.
Kobe, Japan
email: {Yotsumoto.Tappei@zb, Tanigawa.Kozo@zx}.MitsubishiElectric.co.jp

[‡]Graduate School of Engineering,
Tottori University
Tottori, Japan
email: {s112033, takahashi, kawamura, sugahara}@eecs.tottori-u.ac.jp

***Abstract*— A human tracking system based on mobile agent technologies has been proposed to achieve automatic human tracking function. In this system, each target person is tracked automatically by its mobile agent moving among cameras in which a person is detected. The current system utilizes an algorithm to predict which camera detects the target person. The algorithm needs a lot of information from all cameras about their monitoring ranges of the cameras. If one monitoring range is updated by a pan / tilt / zoom operation or some other reason, the whole calculation to determine the relationship between camera nodes must be performed accordingly. In order to solve this problem, we propose in this paper an algorithm that uses only localized node information from each camera and its neighboring camera. With this proposed algorithm, each camera is able to calculate its neighbor nodes without obtaining the monitoring ranges of all cameras. This enables the construction of robust human tracking systems.**

Keywords-Human tracking; Mobile agent; Pan/Tilt/Zoom; Neighbor relation; Localization.

I. INTRODUCTION

In recent years, in order to enhance public security, various kinds of systems such as entrance and exit management and detection of suspicious persons have been introduced. The most widely used system is a supervising system using cameras. In this system, operators must fix their eyes on two or more cameras to find a suspicious person. However, considering the ability of the operators, the maximum number of cameras should be two or three for one operator. A number of operators are required for monitoring many cameras and tracking multiple people. In order to monitor many cameras or track multiple people, employing more operators required. Moreover, when an operator loses sight of a suspicious person, the operator must go over multiple cameras to find the suspicious person. For this reason, systems that enable automatic human tracking using multiple cameras are proposed. These systems, however, have two problems: (A) the computational cost for tracking a person is concentrated on one monitoring server; and (B) loss of tracking ability due to a change in camera monitoring range.

We have proposed an automatic human tracking system based on mobile agent technology. This system consists of cameras, tracking servers, mobile agents, and a monitoring terminal. In this system, a tracking server installed in each camera analyzes images received from the camera. Therefore, the computational cost of image analysis is distributed between each tracking server. A mobile agent is prepared for each person being tracked. The mobile agent migrates among tracking servers by detecting the physical data of a person being tracked. By checking the location of an agent at the monitoring terminal, the operator is able to know the location of the tracked persons.

Tracking all detected persons is possible if a number of cameras that monitor in all directions without blind spots are installed. However, it is an unrealistic idea and is very costly. A more realistic approach is to install a certain numbers of cameras at some specific points such as entrances of a building or rooms and passage crossings. In this case, occasionally a tracked person disappears from a camera's monitoring range. When the human tracking system loses a tracked person, the system has to check every camera's view to find the lost person. A high computation cost for each camera is required for image analysis. Therefore, the algorithm was proposed to predict which camera would display the tracked person next [1].

The algorithm calculates neighbor nodes of each camera based on the value of each camera's monitoring range, map of the floor, and the locations where cameras are installed. A node is defined as a location of a tracking server with a camera. If a tracked person goes out of the monitoring range of one camera, the person should appear in that camera's neighbor nodes. In this case, the algorithm calculates only some nodes, which are from the neighboring camera and the calculation cost for image analysis will be low. Still, there is another case when the monitoring ranges of some camera changes by panning / tilting / zooming operations or other

 ISBN: 978-1-61208-427-5

reasons. In this case, the algorithm will require every camera's current monitoring range, and must re-calculate all the neighbor nodes. However, it is not practical to change the monitoring ranges frequently.

In this paper, the current human tracking algorithm is extended to localize the neighbor node calculation. The proposed algorithm utilizes only the monitoring range of neighbor nodes. Furthermore, this realizes a robust human tracking system that enables continuous tracking even when some nodes are down.

Section II of this paper describes the related research. Section III introduces the proposed human tracking system and describes the neighbor node calculation algorithm. Section IV describes the localized neighbor node calculation algorithm. Section V shows the experimental result, and Section VI contains the conclusion of this paper.

II. RELATED RESEARCH

There are some studies for human tracking between plural cameras.

Y. Shirai researched about tracking multiple persons and proposed a technique for collaboration between cameras for tackling obstruction [2]. N. Kawashima proposed a tracking technique that eliminates noise such as shadow by using a dispersion matrix and by improving the background subtraction method [3]. This research was aimed at accuracy enhancement of persons' detection by using multiple cameras, but was unrelated to track a target person across multiple cameras. Since the image recognition has a possibility that an error occurs, we took an approach that does not rely on image recognition.

H. Mori proposed a tracking technique in an environment where the monitoring ranges of multiple cameras are overlapped by unifying the monitoring images from several cameras [4]. A. Nakazawa proposed a mechanism for combining the physical data of multiple persons [5]. N. Ukita proposed a system to exchange monitoring images efficiently using an agent based framework [6]. This research assumed that the imaging ranges of cameras are overlapped. N. Ukita and D. Makris proposed a method for estimating the migration path of a target based on entry-exit(in-out) information [7][8]. These methods require re-collection of the entry-exit (in-out) information when the monitoring ranges of cameras change. N. Takemura's research predicted possible routes which a person could take from one position and speed of the person [9]. Since the information of the appearance and the moving speed of the person are affected by uncertain movement of the person, we took an approach to predict which camera would display the tracked person next from the information of the equipment (e.g., a monitoring range of camera).

Y. Tanizawa proposed a mobile agent-based framework, called "FollowingSpace" [10]. In this system, when a user moves to another location in a physical space, a mobile agent attached to the user migrates to one of the nearest hosts from the current location of the user. T. Tanaka also proposed an agent-based approach to track a person [11]. However, a mechanism to predict in which camera a target would appear next was not explored. K. Aoki proposed a cooperative surveillance system using active cameras [12]. In this system, each active camera adjusts its observation area to decrease blind spots. P. Ibach et al. proposed an algorithm that employs clustering of mobile nodes [13]. This algorithm is combined with techniques for position based routing. An approach using stochastic locking and semi-hierarchical grouping for a peer-to-peer shared memory system was proposed in [14]. In these studies, a way to cooperate with the node located near is shown. However, we emphasize connection relationship of nodes than their location. Even if distance between nodes is far, it is possible to obtain plural cameras would display the tracked person next by following the connection relationship.

III. HUMAN TRACKING SYSTEM USING MOBILE AGENT TECHNOLOGIES

An automatic human tracking system using mobile agent technologies has been developed [1]. In the system, there are multiple mobile agents, each of which tracks one person called a "target." Since all the targets are tracked automatically by each of the mobile agents, the location of each target can be known by monitoring the location of its corresponding mobile agent.

A. System configuration

The structure of our automatic human tracking system is shown in Figure 1.

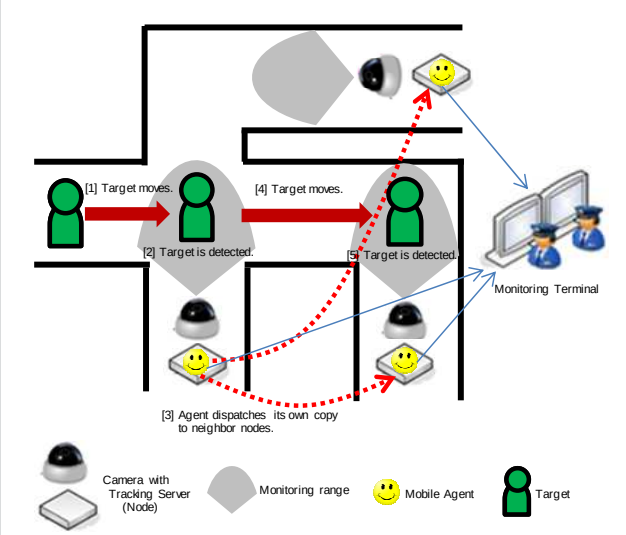


Figure 1. Structure of the proposed system.

The system consists of cameras, tracking servers, mobile agents, and a monitoring terminal. Cameras are discretely installed in a monitoring area and have pan / tilt / zoom functions which change each camera's monitoring range. A tracking server is connected to each camera and receives images from the camera. Tracking servers have the execution environment for a mobile agent and an image analysis

function. Since the image analysis is performed in each tracking server, the computational cost of image analysis is distributed to each tracking server. The mobile agent migrates across tracking servers in accordance with the movement of a target. The locations of mobile agents are displayed in a monitoring terminal. The current positions of all targets can be known through the location of mobile agents.

B. *Tracking flow*

When a target comes into the monitoring range of each tracking server, the tracking server checks whether the target is being tracked by any agent in the server. If it is not, the tracking server generates a mobile agent containing the physical data of the target (e.g., facial features, color of attire). The mobile agent tracks the target based on the target's physical data. At the same moment, the tracking server will distribute copies of the active agent to tracking servers of neighboring cameras located in areas where the target may pass. The calculation algorithm for neighbor nodes is described in the next subsection. Tracking servers of neighboring cameras analyze the camera image periodically based on the physical data in the copy agents to check if the target is in sight. If the target is detected by a tracking server in a neighboring camera, the copy agent of that camera becomes the new active agent and distributes new copies to tracking servers of neighboring cameras. The original active and copy agents are subsequently erased. Besides that, if an agent loses track of the target for a definite period of time, the agent removes itself. The agent exists in the last known position until it is removed.

C. *Algorithm to calculate neighbor nodes*

Regarding camera location, it is practical to install cameras only at specific places, such as building entrances, rooms, or passage crossings. In such an environment, the techniques [4] for tracking a target by using overlapping ranges are not applicable when a target disappears from the monitoring range of a camera. In this case, it becomes necessary to predict which camera a target will appear in next.

In order to predict the next camera, the points that represent a route through which a target can pass should be defined:

- Branch points (passage crossings)
- Camera points (camera locations)
- Viewing points (between two branch points, between two camera points, and between a branch point and a camera point)

The monitoring range of each camera is determined from these points, as shown in Figure 2.

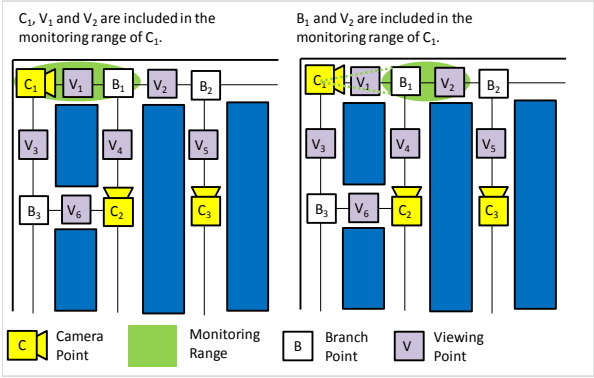


Figure 2. Representation of a route.

The monitoring range of each camera changes by pan, tilt and/or zoom operations. For example, when the monitoring range of camera C_1 is as in the left part of Figure 2, the monitoring range of C_1 becomes $[V_1, B_1]$. When the monitoring range of C_1 is as in the right map of Figure 2, the monitoring range of C_1 becomes $[B_1, V_2]$. Matrix X of $|C| \times |P|$ is defined from the monitoring range of all cameras. Element X_{ij} of matrix X is defined as (1).

$$X_{ij} = \begin{cases} 0, & \text{where the monitoring range of the camera } C_i \text{ does not include the point } P_j. \\ 1, & \text{where the monitoring range of the camera } C_i \text{ includes the point } P_j. \end{cases} \quad (1)$$

Here, C is a set of camera points and P is a set of branch points, camera points and viewing points. Cameras that have overlapping monitoring ranges (neighboring cameras) are identified using (2).

$$D = X \bullet X^T \quad (2)$$

The monitoring range of Camera C_i and C_j are overlapped if $D_{ij} \geq 1$.

Next, the adjacency matrix Y of $|P| \times |P|$ is defined. Element Y_{ij} of matrix Y is defined as (3).

$$Y_{ij} = \begin{cases} 0, & \text{where the point } P_i \text{ and the point } P_j \text{ are not neighboring each other.} \\ 1, & \text{where the point } P_i \text{ and the point } P_j \text{ are neighboring each other.} \end{cases} \quad (3)$$

When $E_{ij} \geq 1$ in (4), the neighboring camera is overlapped with (n-1) points away from the monitoring range of the camera C_i.

$$E = X \bullet Y^n \bullet X^T \quad (4)$$

Even if the neighboring cameras can be identified by (4), the number of points between the monitoring ranges of two cameras is unknown. In other words, n is unknown. Therefore, points which are not included in the monitoring range of all camera from matrix X and Y are eliminated. Matrix X' is generated from matrix X by eliminating all the points in column j that satisfy (5).

$$\sum_{k=i}^{m} X_{kj} = 0 \tag{5}$$

Similary, matrix Y' is generated from matrix Y by eliminating all the points in column j and row j, and by connecting two points i and k if $X_{ij} = 1$ and $X_{jk} = 1$. By directly connecting two points which connected through an eliminated point, it prevents a route from being cut off by the elimination of a point. The next camera can be predicted by calculating (6) from matrix X' and Y'.

$$E' = X' \bullet Y' \bullet X'^{T} \tag{6}$$

IV. LOCALIZATION OF NEIGHBOR NODE CALCULATION

The neighbor nodes can be calculated by the algorithm described in Section III-C. The algorithm, however, needs matrix X that contains the monitoring ranges of all cameras. Next, it is necessary to identify new branch and viewing points due to changes of monitoring ranges as a result of pan / tilt / zoom functions. The more cameras there are in the system, the higher the number of calculation points must increase. Therefore, we localize the algorithm for decreasing the number of points for the calculation. Localization achieves neighbor node calculation without using all points in the system.

In the localized calculation, the neighbor nodes of one camera are calculated by the camera itself. All of the points in the system are not required to calculate neighbor nodes of each camera. Each camera manages only determined points within its monitoring range and those located between its monitoring range and the monitoring range of its neighbor nodes. Let us define a matrix Yc. The elements of Yc are defined as (7).

$$Yc_{ij} = \begin{cases} 0, & \text{where the point } \mathrm{Pc_i} \text{ and the point } \mathrm{Pc_j} \\ & \text{are not neighboring each other.} \\ 1, & \text{where the point } \mathrm{Pc_i} \text{ and the point } \mathrm{Pc_j} \\ & \text{are neighboring each other.} \end{cases} \tag{7}$$

Here, $\mathrm{Pc_i}$ and $\mathrm{Pc_j}$ are the points included in the monitoring range of camera C or the points located between the monitoring range of camera C and the monitoring range of its neighbor nodes. Similarly, we define matrix Xc by (8).

$$Xc_{ij} = \begin{cases} 0, & \text{where the monitoring range of the camera } \mathrm{C_i} \\ & \text{does not include the point } \mathrm{Pc_j}. \\ 1, & \text{where the monitoring range of the camera } \mathrm{C_i} \\ & \text{includes the point } \mathrm{Pc_j}. \end{cases} \tag{8}$$

Xc and Yc consist of a set number of points for each camera. The total number of points in the system does not need to be known. Next, matrix Xc' and Yc' are derived from matrix Xc and Yc using the method described in Section III-C. Then, the neighbor nodes are calculated by:

$$Ec' = Xc' \bullet Yc' \bullet Xc'^{T} \tag{9}$$

The localized calculation achieves a robust human tracking system because all points in the system are not required. If the monitoring range of a camera changes, the localized algorithm requires the updated matrixes of that camera and its neighboring cameras. The following subsection describes update flows when a monitoring range changes.

A. Support for the change of monitoring range

A change of monitoring range of a camera may cause a change of its neighbor node. If that happens, the camera notifies its neighbor cameras that the monitoring range has changed. Cameras that receive this notice update their Xc matrix with the updated monitoring range. Furthermore, in the event that the monitoring range of a camera crosses the monitoring camera of a neighbor node, the matrix Xc and Yc of the camera will not have some points required for calculating neighbor nodes because some new points are generated by the crossing monitoring ranges. For example, in the left part of Figure 3, camera C_1 has monitoring range information for the neighbor node C_2, but does not have the information for C_3. Then points C_1 and V_3 of matrix $Xc1$ become 1. When the monitoring range of C_1 changes as seen in the right part of Figure 3, C_1 and C_3 become neighbors. However, $Xc1$ and $Yc1$ of C_1 have no information about the monitoring range of C_3. Therefore, C_1 gets $Xc2$ and $Yc2$ of C_2, and combines $Xc1$ and $Yc1$ with $Xc2$ and $Yc2$. Then, C_1 can update its neighbor nodes by running a localized neighbor node calculation because $Xc2$ and $Yc2$ include information about points between C_2 and C_3. The points V_2, B_2, V_5 and C_3 on a route to C_3 are added to $Xc1$ and $Yc1$. The point C_1, B_1 and V_2 of matrix $Xc1$ become 1.

 ISBN: 978-1-61208-427-5

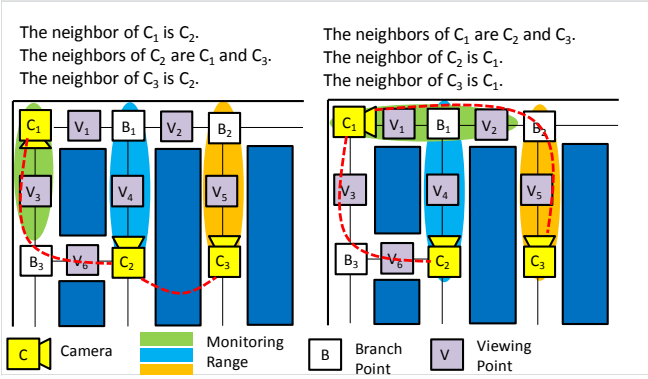


Figure 3. Change of monitoring range.
Dashed line shows that two cameras are neighbors.

B. *Support for additional camera*

If a new camera is added to the system, it will not have matrixes *Xc* and *Yc* . Therefore, when a new camera is installed, we give access information (e.g., IP address and authentication information) about neighbor nodes to the new camera. Hereafter, C_{new} represents a new camera and C_1 represents its neighbor node. C_{new} receives the matrix *Xc*1 and *Yc*1 from C_1. Since C_1 is adjacent to C_{new}, the location of C_{new} installed is included in *Xc*1 and *Yc*1 . Therefore, C_{new} can calculate its neighbor nodes by using *Xc*1 and *Yc*1 .

For example, in the left part of Figure 4, *Xc*1 and *Yc*1 have information about the points from C_1 to C_2, and C_1 to C_3.

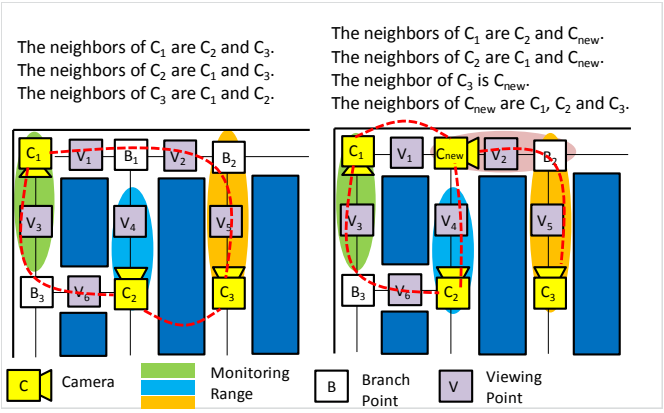


Figure 4. Addition of a camera.
Dashed line shows that two cameras are neighbors.

When C_{new} is installed as seen in the right part of Figure 4, C_{new} receives *Xc*1 and *Yc*1 from C_1. Since *Xc*1 and *Yc*1 contain all the information required to calculate neighbor nodes of C_{new}, C_{new} can calculate its neighbor nodes C_1, C_2 and C_3.

Moreover, the addition of C_{new} is accompanied with the updating of points. Figure 5 shows an example of this update.

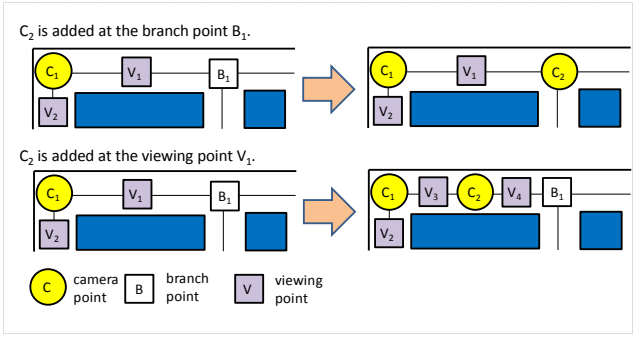


Figure 5. Change of the points.

When a camera is added at a branch point, the branch point is changed to a camera point. When a camera is added at a viewing point, the viewing point becomes a camera point and the viewing point is divided into two points, which are on the both sides of the camera point.

Updated points are added to *Xc*1 and *Yc*1 , and C_{new} updates *Xc*1 with its new monitoring range. The updated matrix *Xc*1 and *Yc*1 are matrixes *Xcnew* and *Ycnew* . Then, C_{new} calculates neighbor nodes using *Xcnew* and *Ycnew* . Then, C_{new} notifies its neighbor nodes of the updated points and its monitoring range. Cameras that receive this notice update their *Xc* and *Yc* matrixes and recalculate their neighbor nodes. In this way, the system can effectively handle the change of neighbor nodes as a result of the addition of a camera.

C. *Support for removing a camera*

As for removing cameras, two cases must be considered; intentional removal and unintentional removal. Because unintentional removal results in the sudden loss of camera C_{rem}'s *Xcrem* and *Ycrem* matrixes, it is more difficult to handle than intentional removal. Therefore, only the case of unintentional removal will be discussed.

To prevent sudden loss of matrixes *Xcrem* and *Ycrem* when unintentional removal occurs, each camera exchanges its *Xc* and *Yc* matrixes with its neighbor node automatically. Each camera always monitors its neighboring cameras to see if they are accessible or not. If any one of the cameras becomes inaccessible, the camera is assumed to have been removed unintentionally. Suppose camera C_1 detects removal of C_{rem}, C_1 combines the *Xcrem* and *Ycrem* with its *Xc*1 and *Yc*1 matrixes. In the combined *Xc*1 matrix, C_1 sets the rows corresponding to the camera point of C_{rem} to 0. This means that the monitoring range of C_{rem} becomes 0. By calculating using the method in Section IV-A, C_1 can calculate its neighbor node even if C_{rem} is intentionally removed. Note that the update of points is not required because unnecessary points (e.g., a camera point corresponding to C_{rem}) are deleted automatically using equation (5) in Section III-C.

 ISBN: 978-1-61208-427-5

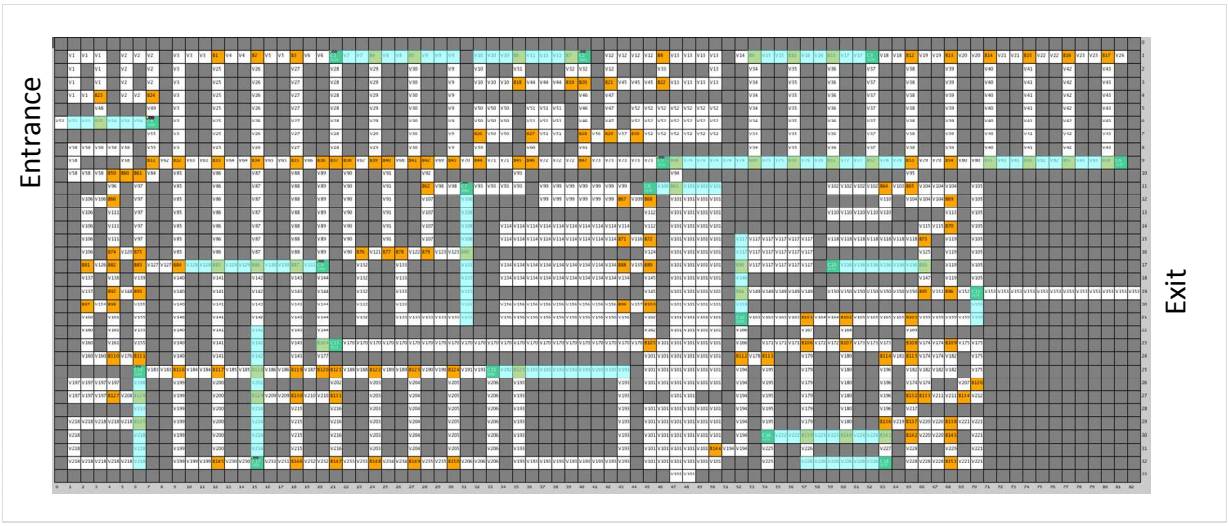


Figure 6. Simulation map.

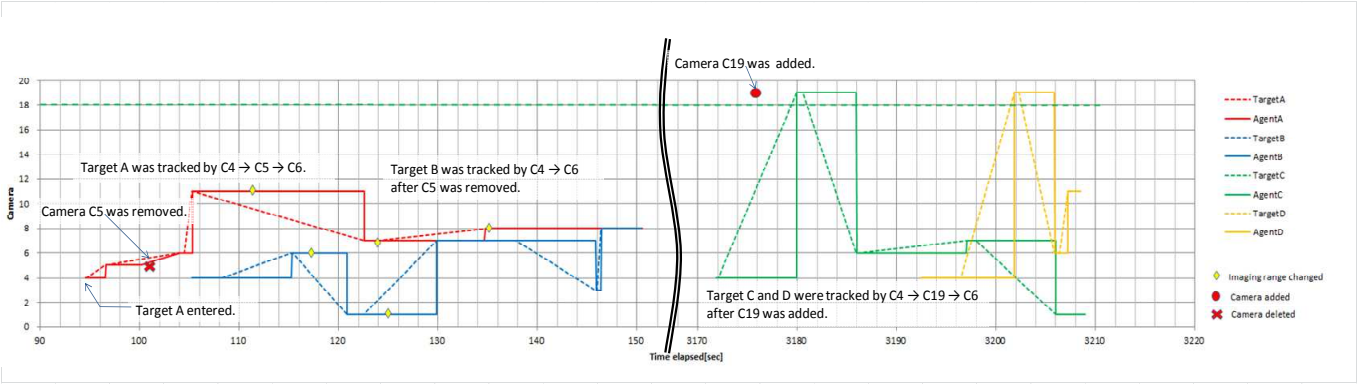


Figure 7. Result of Tracking simulation.

V. EXPERIMENT

The effectiveness of the proposed method was tested using a simulation experiment. The experiment was conducted by installing 18 cameras in 124.5m x 51m area as shown in Figure 6.

Three targets entered the monitoring area from the entrance, walked randomly at a speed of 1.5m/s to 3.0m/s, and then exited the area. When one target exited the area, a new target entered from the entrance. Pan / tilt / zoom of each camera occured randomly once every 30 seconds. Removal and addition of a camera occured once every 8 hours. In this experiment, it was assumed that each camera detects a target accurately because there was a focus on localization of neighbor node calculation. The simulation result is shown in Figure 7. The camera number is displayed on the vertical axis and elapsed time on the horizontal axis. Movements of the targets are shown by dotted lines, and the locations of mobile agents are shown by solid lines. The occurrence times of pan, tilt, and zoom are shown as ◇. Camera addition is shown as ○. Camera removal is shown as ×.

Initially, camera C_5 is adjacent to camera C_4, and C_6 is adjacent to C_5. In Figure 7, target A was detected by C_4, then detected by C_5 and finally detected by C_6. C_5 was deleted after 102 seconds had elapsed. Next, target B was detected by C_4, and then by C_6. This shows that neighbor nodes were correctly updated when C_5 was deleted.

Additionally, camera C_{19} was added between C_4 and C_6 after 3176 seconds had elapsed. Until C_{19} was added, targets were detected by C_6 after being detected by C_4. After C_{19} was added, the target was detected by C_{19} after C_4, and then by C_6 after C_{19}. This means that the neighbor node of C_4 was updated correctly to C_{19}, and that the neighbor node of C_{19} was updated correctly to C_6. Also, even if a monitoring range changed due to pan / tilt / zoom, neighbor nodes were calculated correctly.

The simulation lasted 72 hours. There was no failure in the neighbor node calculation, and targets continued to be tracked by mobile agents accurately.

 ISBN: 978-1-61208-427-5

VI. CONCLUSION

We propose the automatic human tracking system using mobile agent technologies. To track a person using mobile agents, a node that is used to detect a target must first be calculated. The proposed algorithm is able to obtain these neighbor nodes with only localized node information. By using the algorithm, it is possible to calculate neighbor node of each camera without the monitoring ranges of all cameras in the system even when monitoring ranges of cameras are changed or cameras are added / removed. The proposed algorithm provides continuous tracking ability even if some nodes are down. The effectiveness of the proposed system was confirmed in a simulation and experiments. The next step will be large scale experiments using the proposed algorithm to test continuous automatic human tracking in an actual environment.

REFERENCES

[1] K. Tanigawa, T. Yotsumoto, K. Takahashi, T. Kawamura, and K. Sugahara “Determination of neighbor node in consideration of the photographing range of cameras in human tracking system,” The IEICE Transactions on Communications, vol. J97-B, no. 10, Oct.2014, pp. 914-918.

[2] Y. Shirai, and J. Miura, “Human Tracking Complex Environment,” IPSJ Journal. Computer Vision and Image Media, vol. 43, SIG 4(CVIM 4), Jun. 2002, pp. 33–42.

[3] N. Kawashima, N. Nakamura, R. Hagiwara, and H. Hanaizumi, “An Improved Method for Background Sub and Its Application to Tracking of Moving Objects,” IPSJ SIG Technical Report. Computer Vision and Image Media, 2007(87), Sep. 2007, pp. 11–16.

[4] H. Mori, A. Utsumi, J. Ohya, and M. Yachida, “Human Motion Tracking Using Non-synchronous Multiple Observations,” The IEICE Transactions on Information and Systems, vol. J84-D-II, Jan. 2001, pp. 102–110.

[5] A. Nakazawa, S. Hiura, H. Kato, and S. Inokuchi, “Tracking Multi Persons Using Distributed Vision Systems,”IPSJ Journal. vol. 42, no.11, Nov. 2001, pp. 2669–2710.

[6] N. Ukita, “Real-Time Cooperative Multi-Target Tracking by Dense Communication among Active Vision Agent,” The IEICE Transactions on Information and Systems, vol. J88-D-I, Sep. 2005, pp. 1438–1447.

[7] N. Ukita, “Probabilistic-Topological Calibration of Widely Distributed Cameras,” The IEICE Transactions on Information and Systems, vol. J89-D(7), July 2006, pp. 1523-1533.

[8] D. Makris, T. Ellis, and J. Black , ”Bridging the Gaps between Cameras,” CVPR2004, vol.2, 2004, pp. 205-210.

[9] N. Takemura, Y. Nakamura, Y. Matsumoto, and H. Ishiguro, "A Path Planning Method for Human Tracking Agents using Variable-term Prediction", International Conference on Artificial Neural Networks (ICANN), 2010, pp. 407-410.

[10] Y. Tanizawa, I. Satoh, and Y. Anzai, “A User Tracking Mobile Agent Framework “FollowingSpace”,”IPSJ Journal. vol. 43, no. 12, Dec. 2002, pp. 3775–3784.

[11] T. Tanaka, T. Ogawa, S. Numata, T. Itao, M. Tsukamoto, and S. Nishio, “Design and Implementation of a human Tracking System Using Mobile Agents in Camera and Sensor Networks,”IPSJ Transaction on Groupware and Network Services Workshop 2004 , Nov. 2004, pp. 15–20.

[12] K. Aoki, A. Yoshida, S. Arai, N. Ukita, and M. Kidode, ”Functional Assessment of Surveillance of Whole Observation Area by Active Cameras,” IPSJ Journal. vol.48, no.SIG17, 2007, pp. 65-77.

[13] P. Ibach, N. Milanovic, J. Richling, V. Stantchev, A. Wiesner, and M. Malek, "CERO: CE robots community." IEE Proceedings-Software, 152(5), 2005, pp. 210-214.

[14] P. Ibach, V. Stantchev, and C. Keller. "DAEDALUS–A Peer-to-Peer Shared Memory System for Ubiquitous Computing." Euro-Par 2006 Parallel Processing. Springer Berlin Heidelberg, 2006, pp. 961-970.

 ISBN: 978-1-61208-427-5

Implementation of a Generic ICT Risk Model using Graph Databases

Stefan Schiebeck, Martin Latzenhofer,
Brigitte Palensky, Stefan Schauer
Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
e-mail: {stefan.schiebeck.fl | martin.latzenhofer |
brigitte.palensky | stefan.schauer}@ait.ac.at

Gerald Quirchmayr
Research Group Multimedia Information Systems
Faculty of Computer Science
University of Vienna
Vienna, Austria
e-mail: gerald.quirchmayr@univie.ac.at

Thomas Benesch
Research & Development
s-benesch
Vienna Austria
e-mail: thom@s-benesch.com

Johannes Göllner, Christian Meurers, Ingo Mayr
Department of Central Documentation & Information
National Defence Academy of the Austrian Federal
Ministry of Defence and Sports, Vienna, Austria
e-mail: {johannes.goellner | christian.meurers |
ingo.mayr}@bmlvs.gv.at

***Abstract*— Advanced Persistent Threats (APTs) impose an increasing threat on today's information and communication technology (ICT) infrastructure. These highly-sophisticated attacks overcome the typical perimeter protection mechanisms of an organization and generate a large amount of damage. Based on a practical use case of a real-life APT lifecycle, this paper shows how APTs can be tackled using a generic ICT risk analysis framework. Further, it provides details for the implementation of this risk analysis framework using graph databases. The major benefits of this graph database approach, i.e., the simple representation of the interconnected risk model as a graph and the availability of efficient traversals over complex sections of the graph, are illustrated giving several examples.**

Keywords- risk management; APT; ICT security; graph databases; interconnected risk model.

I. INTRODUCTION

Although internal attacks can be seen as today's biggest threat on information security [1], in practice, information security officers still put great emphasis on perimeter control. The internal area of a company's ICT network, e.g., the demilitarized zone (DMZ) or the intranet, is secured based on standard technical guidelines demanding, e.g., the logical separation of a network into subnetworks according to specific security requirements [2]. Nevertheless, the effort invested in monitoring the internal network is moderate. Intrusion detection and prevention systems are cost and time consuming and require a large amount of administration. Recent attack strategies like Advanced Persistent Threats (APTs) take advantage of this lack of internal control.

The term APT summarizes a family of highly sophisticated attacks on an ICT network or infrastructure. Usually, an APT runs over an extended period of time with the objective to steal data and maintain presence indefinitely without being detected. A continuous access allows collecting new data as it emerges, extending the achieved foothold over time, and using the site as a jumping point for the attack on other facilities. The adversary – usually a group of people – has a large amount of resources at hand and applies the whole range of digital, physical and social attack vectors to gain access to a system. The attack is specifically designed for a particular victim, i.e., a company or an organization, such that common security measures can be circumvented effectively. Thus, the adversary stays undetected over a long period of time. One particular technique recurrently used in APTs is social engineering, which exploits the human factor as a major vulnerability of an ICT system. Potential countermeasures, like increasing the staff's awareness concerning ICT security threats via training courses, are not very common. According to a Ponemon study [3], about 52% of the interviewed organizations do not offer respective training courses for their employees.

In the course of the last decade, APTs became one of the most significant kinds of threats on information security, causing a great number of security incidents all over the world [4]. Besides the most prominent APT attack, the application of the malware Stuxnet in an Iranian nuclear power plant, a number of other APT attacks have become known, e.g., Operation Aurora, Shady Rat, Red October or MiniDuke [5][6][7]. As it is shown in the Mandiant Report [4], some adversaries even have a close connection to governmental organizations. The former director of the US cyber command, General Keith Alexander, referred to the currently occurring industrial espionage and theft of intellectual property as "the greatest transfer of wealth in history" [8]. In Europe, the disclosures of Edward Snowden [9] have drawn great attention to this issue. Based on current numbers from cyber-crime reports, which show the growing

 ISBN: 978-1-61208-427-5

amount of damage [10][11], it is distressing how poorly evolved today's countermeasures seem to be.

This paper focuses on the implementation of a generic ICT risk model that can deal with the described issues. The implementation is based on graph databases and social network analysis concepts to provide a perspective that can focus on a specific aspect (node) and its influences (relationships). From a technological perspective, the advantages of the chosen approach are demonstrated, in particular concerning risk aggregation. Therefore, different types of assets, e.g., organizational aspects like processes and personnel, ICT components like IT systems and logical networks, and physical infrastructure objects, serve as examples of assets that are attacked in fictitious, but realistic ways. In detail, after a short overview of related work on graph-based models in Section II, Section III sketches the different steps of an APT attack for a fictitious scenario to illustrate the basic principles of this family of threats. Section IV shows the generic meta-risk-model depicted as a graph model and shortly discusses the pros and cons of an implementation via graph databases vs. relational databases. Section V provides a detailed description of how the generic risk model was implemented using a graph database. Finally, Section VI summarizes the results.

II. RELATED WORK

In general, graph-based models are used to capture relations among system entities at various abstraction levels. In [12], Chartis Research advises the introduction of graph analytics (based on graph databases) to the risk management activities of financial institutions so that they can discover so far unknown risks by revealing interconnected risk patterns. In [13], graph-based representations are applied in the area of risk management for critical infrastructures (CI). Bayesian Networks are used to learn (or simply estimate) CI service risks and their interdependencies. Additionally, a risk prediction is introduced and a case study to validate the model is carried out. However, some of the model's features, like risk prediction and the handling of cyclic dependencies, could not be verified because they simply did not occur during the run-time of the case study. The goal of the approach in [13] is to identify an abstract set of variables and their dependencies based on system measurements. Nevertheless, graph databases have not been used therein. In this paper, we introduce the explicit usage of graph databases (see Section IV for a discussion) to implement an already existing risk scheme retrieved from the IT-Grundschutz framework [14]. With this approach, cascading risks can be represented in a straight-forward way that allows us to run easily through a typical APT attack scenario. The underlying model and functional assessment concept presented in this paper, excluding the usage of a graph database for data manipulation, has been demonstrated in [15], although with the use of a relational database.

III. APT SCENARIO

In [4], the US security company Mandiant describes the typical lifecycle of an APT attack based on an analysis of how a Chinese cyber espionage group infiltrated several companies in the US and worldwide. In the following, the different steps in this APT lifecycle are briefly sketched to give an overview on the basic operations of an APT attack (cf. Figure 1). To provide a better illustration of the scenario, a fictional research facility, *Biomedical Research*, is used. It consists of four research laboratories with increasing degrees of security requirements (*Biosafety Level 1-4*) located in physically separated buildings. Additionally, the research facility runs two data centers, one located in the research building itself, and the other, which works as a backup, located at a distant administrative building. The information most valuable for an attacker is assumed to be hosted in Research Laboratory FL4, which is the one with the highest security level, or in one of the data centers. Based on this setting, a generalized APT attack can be outlined in eight steps.

As a first step, *Initial Recon*, the adversary tries to gain access to the organization's ICT infrastructure. Since the terminals in Research Laboratory FL1 are the only ones having full connection to the internet, a user in FL1 would be a primary target for a spear phishing attack (cf. (1) in Figure 1) in order to place a remote backdoor on either of these terminals. A potential user to be attacked can be identified for example using social engineering. In the second step, *Initial Compromise*, a user in FL1 receives a spear phishing mail and opens the infected attached file (e.g., a ZIP-file). During the execution of the ZIP-file, a basic backdoor (beachhead backdoor, cf. (2) in Figure 1) is installed on the terminal W1. Through this backdoor, a connection to the adversary's command and control server is established. In a third step, *Establish Foothold*, this initial connection is used to install a standard backdoor on the compromised terminal, giving the adversary an increased set of possibilities. Hence, the adversary is able to gain foothold at the application server S1 in FL1 (cf. (4) in Figure 1).

The following four steps (steps 4 to 7) are usually performed more than once, until the adversary acquires the desired information. In step 4, *Escalate Privileges*, the adversary gathers information on valid combinations of user names and passwords inside the internal networks. The attacker also gains additional information about the internal network structure (step 5 – *Internal Recon* – cf. (5) in Figure 1), potentially including internal authentication information. In the following step 6, *Move Laterally*, the adversary infiltrates the local data center as well as the backup data center to locate the valuable information. This is achieved using a vulnerability scan on the file servers S7.1 and S7.2 and an appropriate exploit allowing the compromise of both identically configured systems (cf. (6) in Figure 1). As a final step of this recurrent loop, *Maintain Presence*, all tracks are covered up and the adversary silently stays in the victim's system with an extended foothold (cf. (7) in Figure 1).

The final step, *Complete Mission*, starts when all the target information is collected. Covert channels are established (e.g., using cryptography/steganography) to extract the sensitive information from the file servers (cf. (8) in Figure 1). Afterwards, all traces of the attack are erased.

 ISBN: 978-1-61208-427-5

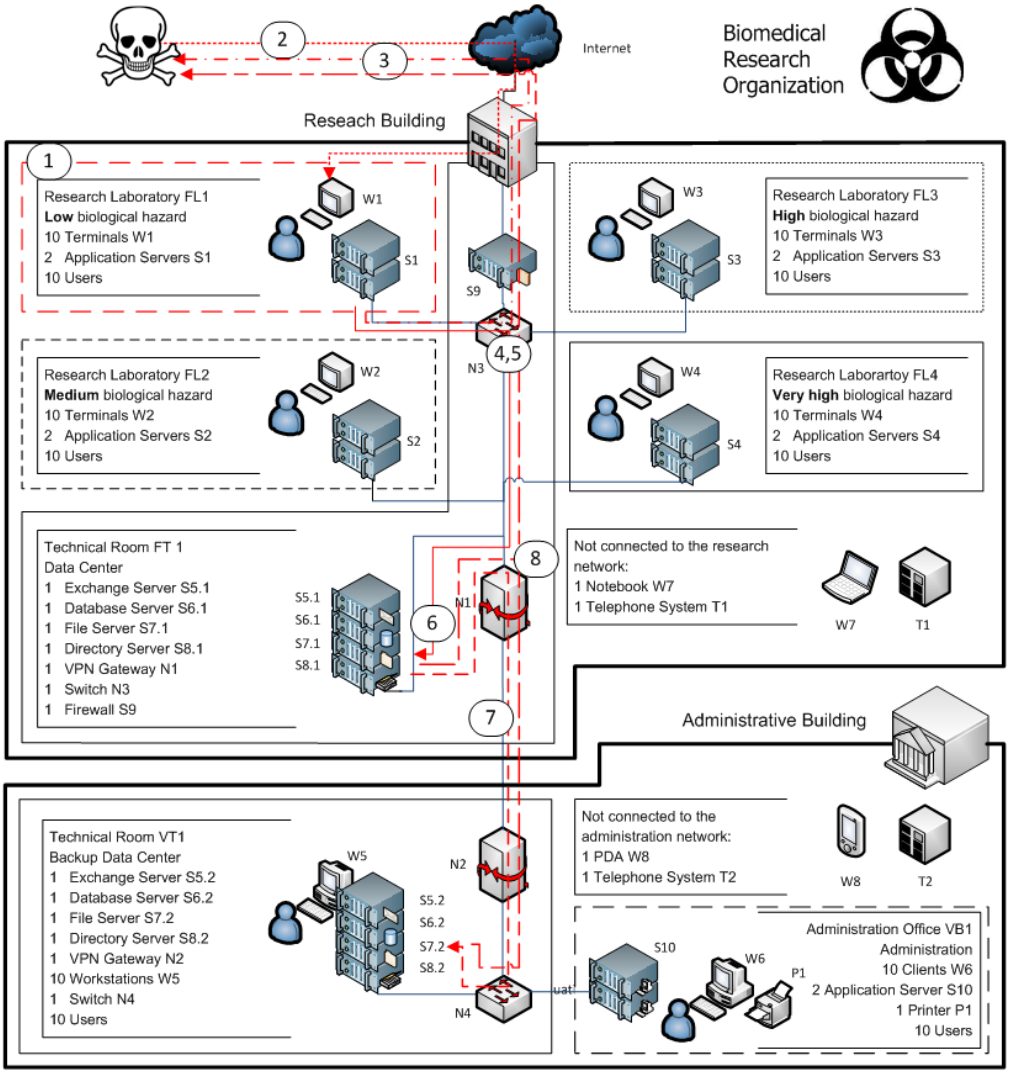


Figure 1 . Sample scenario of a typical APT attack.

IV. GRAPH-BASED META-RISK-MODEL

The meta-model used for risk management depicted in Figure 2 shows the risk graph and its semantic relations. Derived in a combined bottom-up/top-down way, it subsumes all the typical components of common risk management models, tools, processes, and control logic in a generic meta-model (cf. [16] for a more detailed description of the model).

In this work, as architectural background for the implementation of the model the graph database Neo4j [17] is used instead of a relational database. Graph databases provide the advantage of being able to perform near-real-time traversals and aggregations, efficient topology analyses, and the optimal finding of node neighbors [18]. The retrieval time of graph databases is usually significantly less than that of relational databases [19][20]. Moreover, the graph-based implementation ensures more flexibility for defining relationships between datasets. Whereas relational databases are difficult to extend, in graph databases, only a few edges and nodes have to be added to the graph. Thus, the adaption and extension capabilities of the generic meta-model are supported by the schemaless definition of data in graph databases. For instance, additional information on customer and competitor intelligence, responsibilities, or other quantifiable business data can be easily integrated in the database schema. When using Neo4j, the integrated declarative query language CYPHER [17] supports most of the work. Thus, analysis models with extended and adapted functionality are greatly simplified and the graph-based approach is more efficient than business code migration on the software backend.

In situations, where the data set is quite homogenous and rarely changed, other architectural designs, like relational databases or in-memory databases, may be more appropriate. They also offer more support as well as advantages in the field of maturity. Regarding security, MySQL has an extensive security support based on access control lists. In contrast, graph databases like Neo4j expect a trusted environment.

 ISBN: 978-1-61208-427-5

V. PROTOTYPICAL IMPLEMENTATIONS

In the following, we will describe the implementation details of the ICT risk model in graph databases. The underlying approach covers semi-quantitative analysis steps usually used within risk models applying ISO 27005 [21]. We focus on the interconnections in the graph-based meta-model (cf. Figure 2), their representation in the graph database, and the implementation of the APT scenario and a physical attack scenario therein.

A. Graph Model

We propose the following meta-model, illustrated as a graph in Figure 2. Pre-existing information including goals, boundaries, and requirements are narratively documented within the nodes of the risk model. Risk identification is carried out by the definition of organizational assets (*risk model involves asset*) that are depicted by modules (*asset described_by module*), threats (*module threatened_by threats*), safeguards (*threat mitigated_by safeguard*) and roles (*roles responsible_for safeguard*). Goals can be defined based on the usual protection criteria (confidentiality, integrity, availability), as well as on requirements derived from other taxonomies. IT-Grundschutz [14] defines a respective risk catalogue, providing a categorization by module type (applications, IT systems, networks, infrastructure, common aspects), threat type (basic, force majeure, organizational shortcomings, human error, technical failure, deliberate acts), and safeguard type (infrastructure, organization, personnel, hardware and software, communications, contingency planning). Moreover, additional goals and requirements (e.g., stakeholder needs, enterprise goals, IT-related goals, etc.) coming from different frameworks like COBIT [22] can be integrated using cross-references with IT-Grundschutz. The defined goals correlate with the respective exposure of the components within the risk model, which translate to several risk dimensions (*risk model analyses protection criteria*).

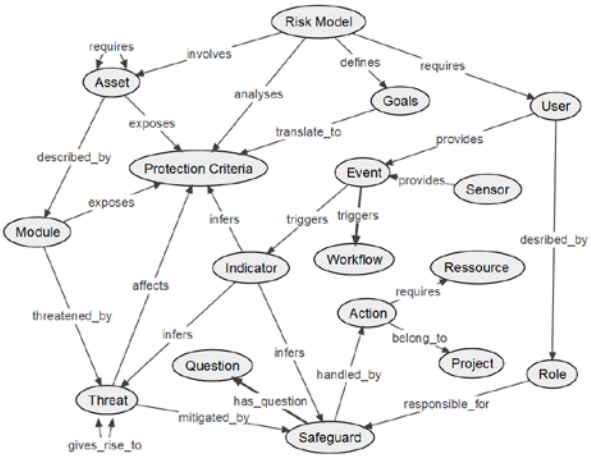


Figure 2 . Graph-based meta-model.

Risk estimation is based on the determination of safeguard maturities (supported by additional control questions, *safeguard has_question question*), threat likelihoods, and impacts on protection criteria. As a result of estimation, exposures are calculated for assets, modules, and threats separately (*asset/module/threat exposes protection criteria*) (cf. Section C).

Assets can optionally be related to each other during scenario analysis in order to depict their dependencies (*asset requires asset*). This supports business impact analysis and the option to perform risk propagation between scenario assets. Another optional step is to perform a detailed threat analysis by modeling threat cascades (*threat gives_rise_to threat*) based on the relationships of the pre-structured scenario model (*Asset requires Asset*).

Users can be associated to the risk model (*risk model requires user*) in specific roles (*user described_by role*) regarding the planning, implementation, and audit of required safeguards (*role responsible_for safeguard*). Users as well as automatable sensors using pre-aggregated data from external support systems (e.g., security information management solutions, i.e., security incident and event management systems (SIEM)) can provide measurements and events to the framework (*user/sensor provides event*), which can be used to trigger workflows (*event triggers workflow*), e.g., when a new IT system is detected.

The framework also provides a possible inference option between objective measurements and related subjective risk factors using fuzzy indicators (*event triggers indicator*) and an expert knowledge system. Inference targets are divided in indicators related to estimated impact, estimated likelihood, and estimated safeguard maturity (*indicator infers protection criteria/threat/ safeguard*).

In order to support basic risk management functionalities, safeguards can be summarized as organizational actions (*safeguard handled_by action*), which can be linked to resources and projects (*action belongs_to*). The intended purpose of the graph model is to provide an easy-to-extend and schemaless model with the ability to interrelate different types of nodes and to aggregate information across affected relationships.

B. Modeling the Scenario in the Graph Database

In the following, the graph-based model of the use case scenario described in Section III is discussed in detail (cf. Figure 3). Assets (blue ovals) are modeled by *_requires_* dependencies, which can be identified by a scenario analysis. The resulting structure defines the top-down inheritance between sub-systems and, at the same time, serves as default path for potential bottom-up threat cascades (*_gives rise to_*). Assets are connected to IT-Grundschutz modules (yellow hexagons) [14], where the referring relation is *described_by*. Threats (red trapezia) are linked to assets by *threatened_by* relations and associated with security measures (green rectangles) by *mitigated_by* relations. For the purpose of a detailed analysis, available threats can be combined to threat cascades via *gives_rise_to* relations. The business impact analysis model (*described_by*) and the IT-Grundschutz taxonomy itself indicate how these cascading paths might look like. This approach of modeling cascades might not address all of the potentially existing correlations, but it provides an easy way of dealing with chained probabilities.

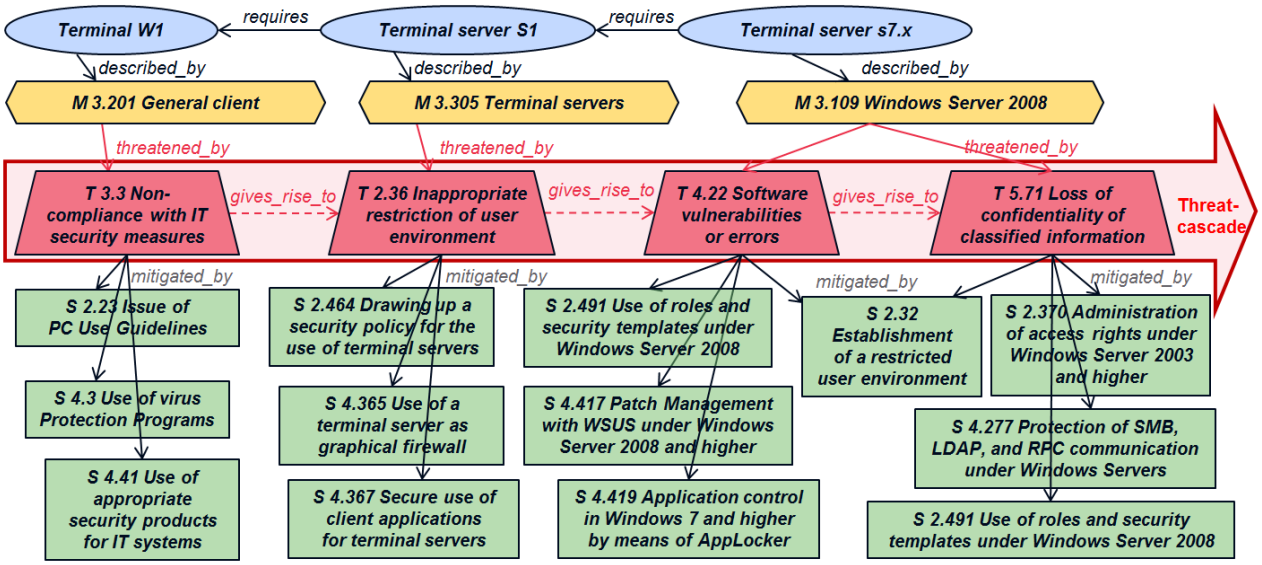


Figure 3 . Graph-based illustration of the scenario.

As described in Section III, at the Terminal W1 (*Module M 3.201 General client*) a user opens a spear phishing mail. This is an exploitation of the organizational threat *T 3.3 Non-compliance with IT security measures*, which is connected with the following security measures:

- *S 2.23 Issue of PC Use Guidelines*
- *S 4.3 Use of virus Protection Programs*
- *S 4.41 Use of appropriate security products for IT systems*

Afterwards, at the corresponding terminal server S1 (*Module M 3.305 Terminal servers*) a standard backdoor is installed. This is possible because of the threat *T 2.36 Inappropriate restriction of user environment*, which could have been addressed by the following security measures:

- *S 2.464 Drawing up a security policy for the use of terminal servers*
- *S 4.365 Use of a terminal server as graphical firewall*
- *S 4.367 Secure use of client applications for terminal servers*

Having gained access to the Terminal server S1, a software vulnerability scan is performed, helping the attacker to exploit the threat *T 4.22 Software vulnerabilities or errors* at the File server S 7.1 and, later on, at the file server S 7.2 (*Module 3.109 Windows Server 2008*). In the analyzed use case, the following security measures were not properly implemented:

- *S 2.32 Establishment of a restricted user environment*
- *S 2.491 Use of roles and security templates under Windows Server 2008*
- *S 4.417 Patch Management with WSUS under Windows Server 2008 and higher*
- *S 4.419 Application control in Windows 7 and higher by means of AppLocker*

At the same module, the follow-up threat *T 5.71 Loss of confidentiality of classified information* can be triggered, which is addressed by the following security measures:

- *S 2.32 Establishment of a restricted user environment*
- *S 2.370 Administration of access rights under Windows Server 2003 and higher*
- *S 2.491 Use of roles and security templates under Windows Server 2008*
- *S 4.277 Protection of SMB, LDAP, and RPC communication under Windows Servers*

In order to perform quantitative analyses, the risk inheritance between different components can be modeled by appropriate functions, e.g., maximum, sum, product, or minimum. More complex normalized, weighted, or bounded variants are also applicable. Possible candidates for the latter are weighted weakest link or prioritized sibling [15][23].

C. Results

In this section, it is demonstrated how the presented risk analysis approach can be used to derive (semi-)quantitative results (e.g. annualized loss expectancy (ALE)) based on semi-quantitative inputs (e.g., safeguard maturity levels according to the Capability Maturity Model Integration (CMMI) framework: 0.. Incomplete, 1.. Initial, 2.. Managed, etc.). In an analog way to the ATP attack example, the analyzed calculation example models the physical environment of an ICT infrastructure. A layered architecture is assumed (cf. Figure 4).

By using graph databases as model environment, the writing of complex business code for risk estimation can be avoided by performing the required assessments using CYPHER statements.

The outlined risk estimation method is a simplified variant of the method defined in [15]. The general view is that vulnerabilities of assets can be exploited by threat sources resulting in negative impacts on protection criteria. Thus, for risk estimation, the vulnerabilities of assets are explicitly taken into account; however, instead of using them directly, they are substituted by maturity gaps of safeguards.

 ISBN: 978-1-61208-427-5

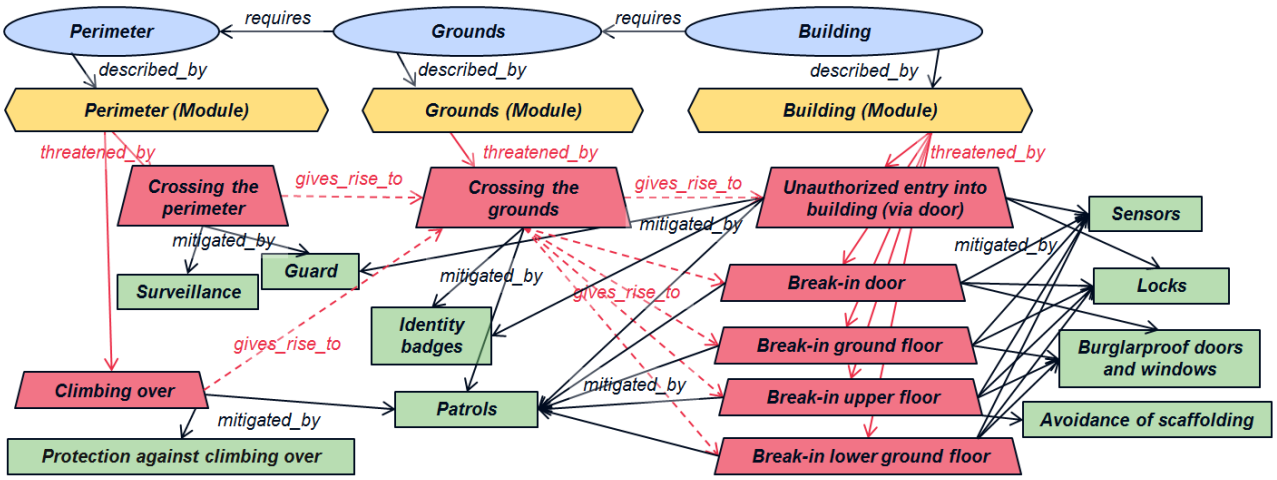


Figure 4 . Graph model for the physical environment scenario (excerpt).

This results in risk as a function of the likelihood of the occurrence of a threat, the maturity gap of an associated safeguard, and the impact that the unwanted event has on protection criteria (cf. (1)).

$$R := f(T_{likelihood}, S_{maturity\ gap}, I_{protection\ criteria}) \quad (1)$$

In an initial step, the safeguard requirements are derived from goals and estimated using maturity levels (from [0…5]). The product of the maturity gap (i.e., 1+maturity gap to evade division by zero) and the safeguard priority (from [1…4]) gives an estimation of the *safeguard exposure* (from [1…24]) (2). Additionally, the relation to the potential maximum exposure (based on the current goal definitions) is also calculated (cf. (3) (4) and Figure 5).

safeguard exposure *= (1+ maturity goal – estimated maturity) * safeguard priority* (2)

safeguard exposure max *= (1+ maturity goal) * safeguard priority* (3)

safeguard exposure % *= safeguard exposure / safeguard exposure max*100* (4)

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:mitigated_by]->(d:USE_CASE:Safeguard)
with (1+r3.target_maturity-r3.maturity)*r3.priority as
  exposure, (1+r3.target_maturity)*r3.priority as
  exposure_max, r3
set r3.exposure = exposure
set r3.exposure_max = exposure_max
set r3.exposure_rel = r3.exposure/r3.exposure_max*100
return r3
```

Figure 5 . Listing for the calculation of safeguard exposures.

After all safeguard exposures are calculated for each asset, the threat likelihoods are estimated for a specific timeframe (from [0…1], however, to simplify the CYPHER code, the null value is excluded to avoid a potential division by zero).

In a next step, the *threat exposures* are calculated. The threat exposure (from [0…20]) depends on the estimated likelihood (from [0…1]) and a function of its safeguard exposures (cf. (5)(6)(7) and Figure 6). For reasons of simplicity, here, the maximum function is used. In order to assess estimation variances, it may be appropriate to estimate the threat likelihood risk-averse (likelihood high) and risk-affine (likelihood low). Based on the calculation of current and potential maximum events, the risk factors within the model can be described either absolutely or relatively.

threat exposure *= likelihood(low) * MAX(safeguard exposure)* (5)

threat exposure max *= likelihood (high) * MAX (safeguard exposure max)* (6)

threat exposure % *= threat exposure / threat exposure max * 100* (7)

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})-
[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:mitigated_by]->(d:USE_CASE:Safeguard)
with max(r3.exposure) as safeguard_exposure,
  max(r3.exposure_max) as safeguard_exposure_max, c
set c.exposure = c.likelihood*safeguard_exposure
set c.exposure_max =
  c.likelihood*safeguard_exposure_max
set c.exposure_rel = c.exposure/c.exposure_max*100
return c
```

Figure 6 . Listing for the calculation of threat exposures.

The threat exposures of all threats that have no incoming *gives_rise_to*-relationships are calculated first. The reason why the exposures of all uninfluenced threats are calculated initially is because no other threats have an effect on them (business impact analysis does not allow cyclic models).

After having calculated the threat exposures of all uninfluenced threats, the threat likelihood of all influenced threats (*gives_rise_to-relations*) can be updated based on the likelihood of their predecessors (chained likelihood). The calculation will be triggered as soon as all predecessors have been calculated. For reasons of simplicity, this is done by a simple multiplication of the original likelihood of the threat and the maximum of the likelihoods of its predecessors. Of course, a more complex function (weighting) representing the relative exposure of the threat to its influences can be

 ISBN: 978-1-61208-427-5

used. In the following example (cf. Figure 7), the originally estimated likelihood of threat *'y'* is multiplied with the maximum of all its incoming *gives_rise_to*-likelihoods.

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:gives_rise_to]->(d:USE_CASE:Threat{name_de:
  'Threat y', module_id:2})
with max(c.likelihood) as trigger_likelihood,
  d.likelihood as original_likelihood, d
set d.original_likelihood = original_likelihood
set d. likelihood = d. likelihood *trigger_ likelihood
return d
```

Figure 7 . Listing for the likelihood update of influenced threats.

After the likelihood update of all threats with incoming *gives_rise_to*-relations is finished, the remaining threat exposures can be calculated.

Depending on the desired level of detail, threats can be assessed individually or as generalized protection criteria related to assets (*asset exposures*), as illustrated by Figure 8. By extending the graph model, arbitrary aggregation layers can be defined. Here, to simplify the outlined use case, asset exposures are aggregated based on the maximum principle and risk is estimated based on the annualized loss expectancy (ALE) formula (cf. (8) and Figure 9). Again, additional lower and upper bounds could be integrated to express variance.

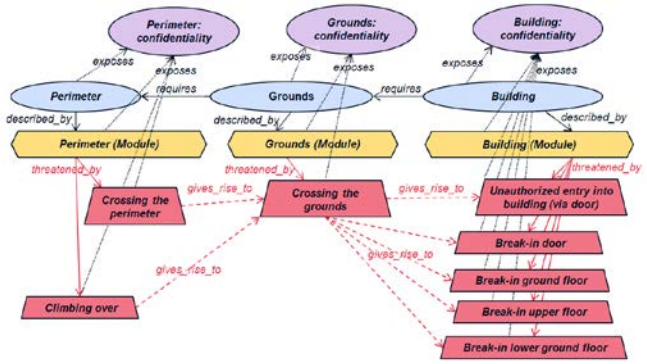


Figure 8 . Possible aggregation of exposures on asset-specific protection criteria (here: confidentiality).

$$\textit{asset risk} = \textit{estimated impact} * \textit{MAX (threat exposure \% / 100)} \quad (8)$$

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de: 'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
with max(c.exposure) as threat_exposure,
  max(c.exposure_max) as threat_exposure_max, a
set a.exposure = threat_exposure
set a.exposure_max = threat_exposure_max
set a.exposure_rel = a.exposure/a.exposure_max*100
set a.ale_risk = a.impact*a.exposure_rel/100
return a
```

Figure 9 . Listing for the calculation of asset exposures and annualized loss expectancy (ALE) risks.

VI. CONCLUSION

This paper describes how a generic meta-model for risk management benefits from the quality of a graph-based implementation, especially from the features of schemaless information which can be parametrized based on the individual requirements of the organization, near-real-time traversals and flexible definitions of relationships between nodes, and the ability of easy model extension. An APT scenario is introduced to demonstrate a practical application of the presented meta-risk-model. The generic nature of the model allows addressing all kinds of threats - from the cyber over the physical to the business realm - and their dependencies. The consideration of cascading risk effects, including human-based information system vulnerabilities, is a necessary prerequisite for an effective defense against APTs, which exploit the full range of attack vectors, from social over digital to physical.

The presented approach shows the application of a combination of several analysis steps and different parts of existing methods, e.g., morphological matrices, fault-tree- and event-tree-analysis, scenario analysis, threat analysis, system decomposition, and functional relationships. The advantages of the presented combined approach are, for example, the possibility to focus on special requirements of information security and to cover a broader range of analysis depth and detail. These features cannot be achieved by using the previously mentioned methods on their own.

The introduced APT scenario is represented as a particular instance of a graph-based implementation of the generic meta-risk-model. The relevant risk components, which can be easily integrated into the graph-based meta-model, are provided by widely-accepted ICT risk frameworks, most importantly by IT-Grundschutz. The defined relations between relevant risk components within this framework give an excellent starting point for possible paths that potential cascading risk effects might take.

From a technical point, for modeling and inference analysis of threat cascades, the graph-oriented database Neo4j with its query language CYPHER was used. Threat cascades and their relations can be visualized by graph databases in a more optimized way compared to relational databases. The schemaless data model of graph databases allows an easier adaption during the modeling process and the application of traversals to integrate calculations without modifications of the business code. However, with regard to the correctness of the results, the domain has to be specified and defined with a low level of uncertainty, and the level of detail of the risk factors has to correlate with the granularity of the results to guarantee a consistent distribution of risk values. Within the discussed use cases, uncertainty resulting from subjective assessments, or inconsistencies and errors in modelling depth is not dealt with explicitly. It can be addressed, like any other aspect, by introducing semi-quantitative descriptors (e.g., assessment uncertainty, etc.), which can be aggregated within the graph model similar to other variables.

 ISBN: 978-1-61208-427-5

ACKNOWLEDGEMENT

The research project ”MetaRisk” (Project-Nr. 840905) is supported and partially funded by the Austrian National Security Research Program KIRAS (http://www.kiras.at/) [23].

REFERENCES

[1] T. W. Coleman, “Cybersecurity Threats Include Employees,” International Policy Digest. [Online]. Available: http://www.internationalpolicydigest.org/2014/05/12/cybersecurity-threats-include-employees/. [Accessed: 19-Mar-2015].

[2] SANS Institute, “Critical Security Controls: Guidelines.” [Online]. Available: http://www.sans.org/critical-security-controls/guidelines. [Accessed: 19-Mar-2015].

[3] Ponemon Institute, “Exposing the Cybersecurity Cracks: A Global Perspective. Part 2: Roadblocks, Refresh and Raising the Human Security IQ,” Traverse City, Michigan, USA, 2014.

[4] Mandiant Intelligence Center, “APT1. Exposing One of China’s Cyber Espionage Units,” Mandiant, Alexandria, Washington, DC, Feb. 2013.

[5] D. Moon, H. Im, J. Lee, and J. Park, “MLDS: Multi-Layer Defense System for Preventing Advanced Persistent Threats,” Symmetry, vol. 6, no. 4, Dec. 2014, pp. 997–1010.

[6] C. Tankard, “Advanced Persistent threats and how to monitor and deter them,” Network Security, vol. 2011, no. 8, Aug. 2011, pp. 16–19.

[7] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, “Combating advanced persistent threats: From network event correlation to incident detection,” Computers & Security, vol. 48, Feb. 2015, pp. 35–57.

[8] The Commission on the Theft of American Intellectual Property, “The IP Commission Report,” National Bureau of Asian Research, May 2013.

[9] G. Greenwald, No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State, Metropolitan Books, 2014.

[10] Internet Crime Complaint Center, “2013 Internet Crime Report,” Federal Bureau of Investigation, 2013.

[11] BMI, “Polizeiliche Kriminalstatistik 2013,” Bundes-ministerium des Innern, Berlin, 2013.

[12] Chartis Research, “Looking for Risk. Applying Graph Analytics to Risk Management. Leading practices from YarcData,” 2013.

[13] T. Schaberreiter, “A Bayesian Network Based On-line Risk Prediction Framework for Interdependent Critical Infrastructures,” Dissertation, University of Oulu, Oulu, Finlande, 2013.

[14] BSI, “IT-Grundschutz-catalogues 13th version 2013,” Bundesamt für Sicherheit in der Informationstechnik - Federal Office for Information Security, Bonn, Germany, 2013.

[15] S. Schiebeck, “An Approach to Continuous Information Security Risk Assessment focused on Security Measurements,” Dissertation, University of Vienna, Wien, 2014.

[16] J. Göllner, T. Benesch, S. Schauer, K. Schuch, S. Schiebeck, G. Quirchmayr, M. Latzenhofer, and A. Peer, “Framework for a Generic Meta Organisational Model,” paper presentation at the 14th FRAP - Finance, Risk and Accounting Perspectives Conference, Oxford, United Kingdom, 2014.

[17] Neo4j Graph Database, “Intro to Cypher - Neo4j Graph Database.” [Online]. Available: http://neo4j.com/developer/cypher-query-language/. [Accessed: 25-Mar-2015].

[18] R.-G. Urma and A. Mycroft, “Source-code queries with graph databases—with application to programming language usage and evolution,” Science of Computer Programming, vol. 97, Jan. 2015, pp. 127–134.

[19] C. Batra and C. Tyagi, “Comparative Analysis of Relational And Graph Databases,” International Journal of Soft Computing and Engineering, vol. 2, no. 2, May 2012, pp. 509–512.

[20] C. T. Have and L. J. Jensen, “Are graph databases ready for bioinformatics?” Bioinformatics, vol. 29, no. 24, Dec. 2013, pp. 3107–3108.

[21] ISO International Organizaion of Standardization, ISO/IEC 27005 - Information technology -- Security techniques -- Information security risk management, 2011.

[22] ISACA, “COBIT 5 - Enabling Processes,” Rolling Meadows, Illinois, 2012.

[23] C. Wang and W. A. Wulf, “Towards a Framework for Security Measurement,” in Proc. of 20th National Information Systems Security Conference, Baltimore, Maryland, 1997, pp. 522-533.

 ISBN: 978-1-61208-427-5

Reduction of Neighbor Node Calculations for Automatic Human Tracking System

Miki Tsuji†, Tappei Yotsumoto†‡, Kenichi Takahashi†,
Kozo Tanigawa‡, Takao Kawamura†, and Kazunori Sugahara†

†Graduate School of Engineering, Tottori University, Tottori, Japan
email: {s112033, takahashi, kawamura, sugahara}@eecs.tottori-u.ac.jp

‡System Engineering Department, Melco Power Systems Co.Ltd, Kobe, Japan
email: {Yotsumoto.Tappei@zb, Tanigawa.Kozo@zx}.MitsubishiElectnc.co.jp

Abstract— **Construction method of human tracking systems by using mobile agent technologies is described in this paper. The human tracking system by mobile agent technologies achieves automatic human tracking functions by migrations of mobile agents among camera nodes according with the movements of target persons. The authors have developed the method to decide the destination nodes of mobile agents by calculating neighbor cameras that will catch the tracked person next. In this paper, the method for reducing the amount of calculation to decide the destination of the agent is proposed. In the proposed method, only the data of neighboring cameras which are selected by localization is required. By utilizing this method, it is possible to realize highly robust human tracking systems without having the information of all cameras in the systems.**

Keywords- Human tracking; Mobile agent; Pan / Tilt / Zoon; Neighbor relation; Localization.

I. INTRODUCTION

In recent years, in order to protect security of our society, various kinds of systems, such as entrance and exit management and discovery of trespasser, were introduced. The most widely used one is a supervising system using cameras. In the supervising system using cameras, operators must fix their eyes on a number of cameras to find a suspicious person. However, considering the ability of the operators, the maximum number of cameras should be limited. In case of tracking multiple person by more than one monitor equipment, some operators should cooperate with each other for monitoring. Moreover, when an operator loses sight of a suspicious person, the operator must go over multiple cameras to find the suspicious person. Considering these points, systems which enable to track a person automatically among multiple cameras are proposed.

Until now, various kinds of human tracking system have been proposed to reduce the load of the operators. Shiroi [2] surveys researches tracking multiple persons and proposes a technique to make multiple cameras cooperate with one another.. Kawashima [3] proposes the method to raise the detection accuracy of tracking persons with many cameras that eliminates noises such as shadow by using a dispersion matrix and the background subtraction. These researches aim at improvement of persons' detection accuracy by using multiple cameras, and do not consider the case of tracking targets across multiple cameras.

As the research for tracking across multiple cameras, Mori [4] proposes the tracking technique to unify the monitoring image of multiple cameras. Nakazawa [5] proposes the mechanism for combining feature information of multiple persons. Ukita [6] proposes the system to exchange the monitoring images efficiently by an agent-based framework. Aoki [7] proposes a cooperative surveillance system which realizes surveillance of a whole monitoring area by using active cameras. In this system, each active camera adjusts its monitoring range to decrease blind spots by overlapping with the monitoring range of other active cameras. These researches supposes that the monitoring range of cameras are overlapped.

Some mobile agent-based frameworks have been proposed. Tanizawa [8] proposes a mobile agent-based framework "Following Space". In this system, when a user moves to another location in a physical space, a mobile agent attached to the user migrates to one of the nearest hosts from the current location of the user. Tanaka [9] also proposes an agent-based approach to track a person. However, in these proposals, a mechanism to predict which camera will track a target next is not explained enough.

We proposed a mobile agent-based system for automatic human tracking. Here, the codes which can migrate among tracking servers are called mobile agents. The feature data of a target person that help distinguish him/her from other persons is extracted from camera images in tracking servers and is stored in the mobile agent. Each agent achieves human tracking based on this feature data. The agents migrate among nodes only with small size feature data and it is not necessary to send camera images. For this reason, the amount of communication traffics of proposed system is very small. This system consists of cameras, tracking servers, mobile agents, and a monitoring terminal. In our system, a tracking server installed in each camera analyzes images received from the camera. A mobile agent is prepared for each person being tracked. A mobile agent

 ISBN: 978-1-61208-427-5

migrates among tracking servers by detecting the feature data of a target person. The locations of mobile agents are displayed on a monitoring terminal. The operator is able to know the current locations of all the tracked persons by checking the locations of the mobile agents.

In our system, a certain number of cameras are installed at some specific points such as entrances of a building, of rooms and passage crossing. In such an environment, it sometimes happens that a tracked person disappears from any camera's monitoring view. Therefore, we propose the algorithm [1] to predict camera's view in which the tracked person will appear in next. The algorithm calculates neighbor nodes of each camera based on the value of each camera's monitoring range, the map of the floor and the locations where cameras are installed. Hereafter, term *node* means tracking server and/or camera. By using the algorithm, our system predicts cameras which will catch the tracked person next. However, the algorithm needs every camera's current monitoring range for re-calculation of its neighbor nodes. It is difficult when many cameras change their monitoring ranges frequently by pan / tilt / zoom operation. In this paper, we extend the human tracking algorithm for localizing the neighbor node calculation. By using the localization of neighbor node calculation, re-calculation cost of the neighbor node will be low even when many cameras change their monitoring ranges.

In the following sections, Section II introduces our human tracking system and describes the neighbor node calculation algorithm. Section III describes about the localization of neighbor node calculation. Section IV shows the experimental results, and we conclude the paper in Section V.

II. HUMAN TRACKING SYSTEM UTILIZING MOBILE AGENTS

An automatic human tracking system using mobile agent technologies has been developed. In the system, there are multiple mobile agents, each of which tracks one person called "a target". Since all the targets are tracked automatically by each mobile agent, the location of each target can be known by monitoring the location of its corresponding mobile agent.

A. System configuration

The structure of the system is shown in Figure 1. The system consists of cameras, tracking servers, mobile agents, and a monitoring terminal. Cameras are discretely installed in a monitoring area and have pan / tilt / zoom functions which change each camera's monitoring range.

A tracking server is connected to each camera and receives images from the camera. Tracking servers have also the execution environment for a mobile agent and image analysis function. Since the image analysis is performed in each tracking server, the computational cost of image analysis is distributed to each tracking server.

The mobile agent migrates across tracking server in accordance with the movement of a target. The locations of mobile agents are displayed in a monitoring terminal. The current positions of all targets can be known through the location of mobile agents.

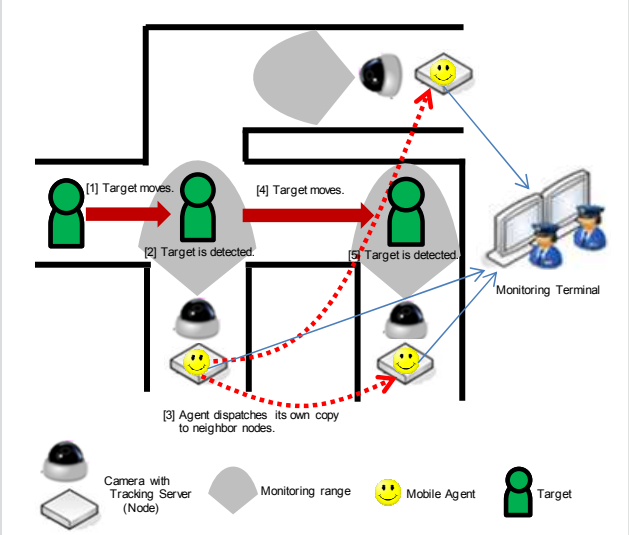


Figure 1. Structure of the proposed system.

B. Tracking flow

When a target comes into the monitoring range of a certain tracking server, the tracking server checks whether an agent tracking the target exists in the server. If it is not, the tracking server generates a mobile agent containing the physical data of the target (e.g., facial features, color of attire). The mobile agent tracks the target based on the target's physical data. At the same moment to the tracking servers will distribute copies of the active agent, that tracking servers of neighboring cameras located in areas where the target may pass. The calculation algorithm for neighbor nodes is described in the next subsection. Tracking servers of neighboring cameras analyze the camera image periodically based on the physical data in the copy agents to check if the target is in sight. If the target is detected by a tracking server in a neighboring camera, the copy agent of that camera becomes the new active agent and distributes new copies to tracking servers of neighboring cameras. The original active and copy agents are subsequently erased.

C. Algorithm to calculate neighbor nodes

Regarding camera location, it is practical to install cameras only at specific places, such as building entrances, rooms, or passage crossings. In such an environment, the techniques [4][5][6][7] for tracking a target by using

 ISBN: 978-1-61208-427-5

overlapping ranges are not applicable. Thus, it is necessary to predict which camera will catch the target next.

In order to predict the next camera, we first define the points to represent a route on which a target can move:

- Branch point which is a passage crossing,
- Camera point where a camera is installed, and
- Viewing point which is prepared between two branch points, between two camera points, and between a branch point and a camera point.

The monitoring range of each camera is determined from these points as shown in Figure 2.

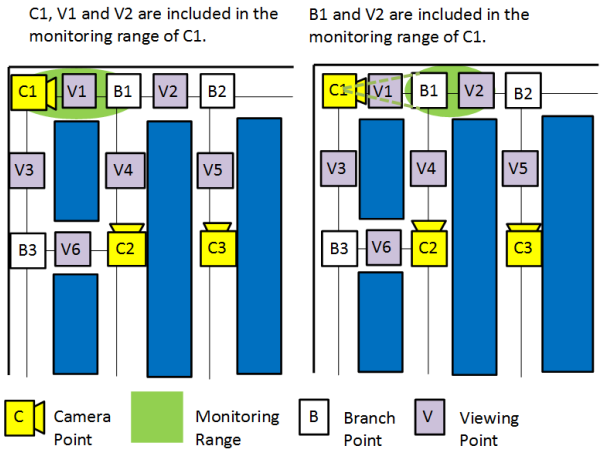


Figure 2. Photographing ranges of camera C1.

The monitoring range of each camera changes by pan, tilt and/or zoom. For example, when the monitoring range of camera C1 is as in the left map of Figure 2, the monitoring range of C1 becomes [V1, B1]. When the monitoring range of C1 is as in the right map of Figure 2, the monitoring range of C1 becomes [B1, V2]. We define matrix X of $|C| \times |P|$ from the monitoring range of all cameras. Here, C is a set of camera points and P is a set of branch, camera, and viewing points. Element X_{ij} of matrix X is defined as (1).

$$X_{ij} = \begin{cases} 0, & \text{In the case where the monitoring range of the camera } C_i \text{ does not include the point } P_j. \\ 1, & \text{In the case where the monitoring range of the camera } C_i \text{ does include the point } P_j. \end{cases} \quad (1)$$

Then, cameras whose monitoring ranges overlap each other are calculated by (2).

$$D = X \bullet X^T \quad (2)$$

The monitoring ranges of camera C_i and C_j are overlapped if $D_{ij} \geq 1$. Next, we define adjacency matrix Y of $|P| \times |P|$. Element Y_{ij} of matrix Y is defined as (3).

$$Y_{ij} = \begin{cases} 0, & \text{In the case where the point } P_i \text{ and the point } P_j \text{ are not neighboring each other.} \\ 1, & \text{In the case where the point } P_i \text{ and the point } P_j \text{ are neighboring each other.} \end{cases} \quad (3)$$

When $E_{ij} \geq 1$ in (4), the monitoring range of the camera C_j overlaps with *(n-1)* points away from the monitoring range of the camera C_i.

$$E = X \bullet Y^n \bullet X^T \quad (4)$$

III. LOCALIZATION OF NEIGHBOR NODE CALCULATION

The neighbor nodes can be calculated by the algorithm described in Section II-C. Matrices X and Y are composed of all points. Therefore, the larger the monitoring area is and/or the larger number of the cameras increases, the larger X and Y become. Therefore, we localize the calculation of neighbor nodes. Localization allows calculation of the neighbor nodes from limited points without considering all points in the system.

A. Localization of matrix X and Y

In the localized algorithm, each camera manages the points included in the monitoring range of the camera and the points located between the monitoring range of the camera and that of its neighbor nodes. Therefore, we define a matrix Yc. The elements of Yc camera C manages are

$$Yc_{ij} = \begin{cases} 0, & \text{In the case where the point } Pc_i \text{ and the point } Pc_j \text{ are not neighboring each other.} \\ 1, & \text{In the case where the point } Pc_i \text{ and the point } Pc_j \text{ are neighboring each other.} \end{cases} \quad (5)$$

Here, Pc_i and Pc_j are the points included in the monitoring range of camera C or the points located between the monitoring range of camera C and the monitoring range of its neighbor nodes. Similarly, we define matrix Xc by (6).

$$Xc_{ij} = \begin{cases} 0, & \text{In the case where the monitoring range of the camera } C_i \text{ does not include the point } Pc_j. \\ 1, & \text{In the case where the monitoring range of the camera } C_i \text{ does include the point } Pc_j. \end{cases} \quad (6)$$

Since Xc and Yc consist of limited points depending on each camera, each camera does not need to manage all points in the system any more.

B. Calculation of neighbor nodes

From the matrices Xc and Yc, we calculate

 ISBN: 978-1-61208-427-5

$$Ec = Xc \bullet Yc^{n} \bullet Xc^{T} \quad (7)$$

When $Ec_{ij} \geq 1$ in (7), the monitoring range of the camera C_j overlaps with *(n-1)* points away from the monitoring range of the camera C_i. Since *Xc* and *Yc* consist of points around the camera, the system does not need to collect every camera's current monitoring range. However, we do not know a number of points between the monitoring ranges of two cameras; in other words, we do not know *n*. Therefore, we eliminate points which are not included in the monitoring range of all cameras from matrix *Xc* and *Yc*. Matrix *Xc'* is generated from matrix *Xc* by eliminating all the column *j* which satisfy (8).

$$\sum_{k=i}^{m} Xc_{kj} = 0 \quad (8)$$

Matrix *Yc'* is also generated from matrix *Yc* by eliminating all the column *j* and row *j* which satisfy (8). At this time, two points connected through an eliminated point should be connected, which prevents a route from being cut off by elimination of a point. Therefore, Xc'_{ik} is set to 1, if $Xc_{ij} = 1$ and $Xc_{jk} = 1$. Then, the next camera is found by calculating (9) from matrix *Xc'* and *Yc'*.

$$Ec' = Xc' \bullet Yc' \bullet Xc'^{T} \quad (9)$$

C. An example of localization

We show an example of localization by using a map in Figure 3. In Figure 3, *Cx* represents a camera point. *Bx* represents a branch point, and *Vx* represents a viewing point. There are 5 camera points, 4 branch points and 10 viewing points in the map. Since the total number of the points is 19, the size of matrix *X* becomes 5 × 19 (|*the number of camera points*| × |*the total number of the all points*|), and the size of matrix *Y* becomes 19 × 19. The monitoring range of each camera is represented by a triangular range.

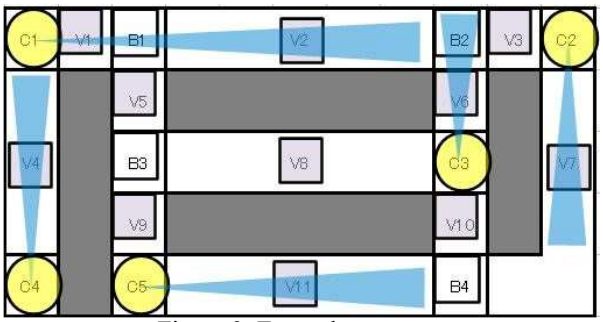


Figure 3. Example map.

Here, we focus on camera C1. Yc_1 consists of points included in the monitoring range of camera C1 and points located between the monitoring range of C1 and its neighbor nodes. Therefore, Yc_1 consists of C1, C3, C5, B1, B2, B3, V1, V2, V4, V5, V8 and V9. Thus, the size of matrix Yc_1 becomes 12 × 12 as shown in (10).

$$Yc_1 = \begin{array}{c} \\ C1 \\ C3 \\ C5 \\ B1 \\ B2 \\ B3 \\ V1 \\ V2 \\ V4 \\ V5 \\ V8 \\ V9 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} C1 & C3 & C5 & B1 & B2 & B3 & V1 & V2 & V4 & V5 & V8 & V9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad (10)$$

Similarly, the size of Xc_1 becomes 4 × 12 as shown in (11).

$$Xc_1 = \begin{array}{c} \\ C1 \\ C3 \\ C4 \\ C5 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} C1 & C3 & C5 & B1 & B2 & B3 & V1 & V2 & V4 & V5 & V8 & V9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad (11)$$

Then, matrix Xc_1' is generated from Xc_1 by eliminating column B3, V5, V8 and V9 because their columns satisfy (8). Thus, Xc_1' becomes 4 × 8 matrix as shown in (12).

$$Xc_1' = \begin{array}{c} \\ C1 \\ C3 \\ C4 \\ C5 \end{array} \begin{array}{c} \begin{array}{cccccccc} C1 & C3 & C5 & B1 & B2 & V1 & V2 & V4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad (12)$$

For the generation of matrix Yc_1', at first, columns and rows of B3, V5, V8 and V9 are deleted from Yc_1. After that, B3, V5, V8 and V9, which satisfy $Yc_{ik} = 1$ and $Yc_{kj} = 1$ is set to 1 in Yc_1'. For example, the values of [i, j] = [B1, B3] and [B3, B1] are set to 1 by the deletion of V5. This prevents cutting off the route between B1 and B3. Similarly, the values of [i, j] = [B1, V8] , [V8, V9], [B1, V9] and [V9, B1] are set to 1 by the deletion of B3; [B1, C3], [C3, B1], [C3, V9]and [V9, C3] is set to 1 by V8; [B1, C5], [C5, B1], [C3,

C5]and [C5, C3] is set to 1 by V9. Thus, the matrix Yc_1' is generated as shown in (13).

$$Yc_1' = \begin{array}{c} \\ C1 \\ C3 \\ C5 \\ B1 \\ B2 \\ V1 \\ V2 \\ V4 \end{array} \begin{array}{c} \begin{array}{cccccccc} C1 & C3 & C5 & B1 & B2 & V1 & V2 & V4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad (13)$$

IV. EXPERIMENT

To confirm effectiveness of the proposed method, the simulation experiment using Java is conducted. In the experiment, we prepared seven maps from 30 × 30 [m^2] to 210 × 210 [m^2]. The number of points in each map is shown in Table I.

Table I. NUMBER OF POINTS IN THE SIMULATION MAP

Map Size [m^2]	Number of points			
	Camera points	Branch points	Viewing points	total
30× 30	8	18	44	70
60× 60	32	80	184	296
90× 90	72	189	418	679
120× 120	128	328	740	1196
150× 150	200	518	1160	1878
180× 180	288	751	1674	2713
210× 210	392	1030	2286	3708

Three targets enter the monitoring area from the entrance, walk randomly at a speed of 1.5 [m/s] to 3.0 [m/s], and go out from the exit. When a target leaves the monitoring area, a new target enters from the entrance. Pan / tilt / zoom of each camera occur irregularly in the ratio once in 30 seconds. Deletion and addition of cameras occur in the ratio once in 8 hours. Table II shows the number of points in *Xc* and *Yc*.

Table II. NUMBER OF POINTS

Map size [m^2]	# of camera points managed in *Xc*			# of all points managed in *Xc*/*Yc*		
	AVE	MAX	MIN	AVE	MAX	MIN
30× 30	6	8	5	34	35	52
60× 60	12	20	5	75	138	25
90× 90	12	26	4	77	162	25
120× 120	16	33	3	99	213	19
150× 150	16	40	3	100	262	17
180× 180	16	41	4	95	256	21
210× 210	16	53	4	95	330	25

The size of matrix *X* becomes |*camera points*|× |*total points*| shown in Table I, and the size of matrix *Y* becomes |*total points*|× |*total points*|. For example, in 120 × 120 [m^2] map, the size of *X* is 128 × 1196, and the size of *Y* is 1196 × 1196. When we apply the localization, the average size of matrix which each camera manages is dramatically reduced in all maps. For example, in 120 × 120 [m^2] map, the average size of *Xc* is 16 × 99, the average size of *Yc* is 99 × 99 shown in Table II. In the way, comparing the Table I and Table II, the number of points having each camera is reduced. This means the change of monitoring range of a camera effects on limited nodes. According to the size of the monitoring area, the number of the points which is expected to be reduced becomes large at accelerated pace.

We also measured the processing time to calculate neighbor nodes. The results are summarized in Table III.

Table III. CALCULATION TIME

Map size [m^2]	Average with localization [ms]	Average without localization [ms]
30× 30	0.054	0.192
60× 60	0.064	0.735
90× 90	0.091	9.268
120× 120	0.128	54.815
150× 150	0.117	216.471
180× 180	0.130	1299.056
210× 210	0.192	7496.471

If the monitoring range of the camera is changed by pan/tilt/zoom, the re-calculation of the neighbor node is needed by (4) each time. We show the calculation time for several map sizes in Table III. We can see in Table III that without localization, when the monitoring area is large, the re-calculation of the neighbor time is very large. In contrast,

 ISBN: 978-1-61208-427-5

with localization, the re-calculation of the neighbor time is not larger than without localization. For example, when we do not apply the localization, the system spends 7496 [ms] for neighbor node calculations in 210 × 210 [m^2] map. In this experiment, since pan / tilt / zoom of each camera occurs irregularly in the ratio once in 30 second, which is already overloaded, the system fails to track targets. On the other hand, when we apply the localization in 210 × 210 [m^2] map, the system can calculate neighbor nodes in 0.192 [ms] in average. That is, the localization enables the system to calculate neighbor nodes even when the size of the map is bigger. In addition to that, the calculation time is almost the same.

V. CONCLUSION

The automatic human tracking system using mobile agent technologies is proposed. For tracking a person by using mobile agent technologies, we need to find the neighbor node which catches the target person. Therefore, we propose the localization algorithm to calculate neighbor nodes. This realizes continuous tracking abilities even if the monitoring ranges of the cameras frequently change. The effectiveness of the proposed system was confirmed in the simulation. The simulation result shows the calculation cost of neighbor nodes is suppressed even when the number of cameras installed in a system increases. The future task is to conduct large scale experiments using the proposed algorithm in an actual environment.

REFERENCES

[1] K. Tanigawa, T. Yotsumoto, K. Takahashi, T. Kawamura, and K. Sugahara, "Determination of neighbor node in consideration of the photographing range of cameras in human tracking system," The IEICE Transactions on Communications, Vol.J97-B, No.10, Oct.2014, pp.914-918.

[2] Y. Shirai, and J. Miura, "Human Tracking Complex Environment," ISPJ Journal. Computer Vision and Image Media, vol. 43, SIG 4(CVIM 4), Jun. 2002, pp. 33–42.

[3] N. Kawashima, N, Nakamura, R. Hagiwara, and H. Hanaizumi, "An Improved Method for Background Sub and Its Application to Tracking of Moving Objects," IPSJ SIG Technical Report. Computer Vision and Image Media, 2007(87), Sep. 2007, pp. 11–16.

[4] H. Mori, A. Utsumi, J. Ohya, and M. Yachida, "Human Motion Tracking Using Non-synchronous Multiple Observations,"The IEICE Transactions on Information and Systems, Vol. J84-D-II, Jan. 2001 , pp. 102–110.

[5] A. Nakazawa, S. Hiura, H. Kato, and S. Inokuchi, "Tracking Multi Persons Using Distributed Vision Systems, "IPSJ Journal. vol. 42, No.11, Nov. 2001, pp. 2669–2710.

[6] N. Ukita, "Real-Time Cooperative Multi-Target Tracking by Dense Communication among Active Vision Agent," The IEICE Transactions on Information and Systems, Vol. J88-D-I, Sep. 2005, pp. 1438–1447.

[7] K. Aoki, A. Yoshida, S. Arai, N. Ukita, and M. Kidode, "Functional Assessment of Surveillance of Whole Observation Area by Active Cameras," ISPJ Journal. Vol.48, No.SIG17, 2007, pp.65-77.

[8] Y. Tanizawa, I. Satoh, and Y. Anzai, "A User Tracking Mobile Agent Framework "FollowingSpace","IPSJ Journal. vol. 43, No.12, Dec. 2002, pp. 3775–3784.

[9] T. Tanaka, T. Ogawa, S. Numata, T. Itao, M. Tsukamoto, and S. Nishio, "Design and Implementation of a human Tracking System Using Mobile Agents in Camera and Sensor Networks, "IPSJ Transaction on Groupware and Network Services Workshop 2004. , Nov. 2004, pp. 15–20.

 ISBN: 978-1-61208-427-5

Overview on Security Approaches in Intelligent Transportation Systems

Searching for hybrid trust establishment solutions for VANETs

Christoph Ponikwar, Hans-Joachim Hof

MuSe - Munich IT Security Research Group
Department of Computer Science and Mathematics
Munich University of Applied Sciences (MUAS), Germany
e-mail: christoph.ponikwar@hm.edu, hof@hm.edu

***Abstract*—Major standardization bodies developed and designed systems that should be used in vehicular ad-hoc networks. The Institute of Electrical and Electronics Engineers (IEEE) in America designed the wireless access in vehicular environments (WAVE) system. The European Telecommunications Standards Institute (ETSI) did come up with the "ITS-G5" system. Those Vehicular Ad-hoc Networks (VANETs) are the basis for Intelligent Transportation Systems (ITSs). They aim to efficiently communicate and provide benefits to people, ranging from improved safety to convenience. But different design and architectural choices lead to different network properties, especially security properties that are fundamentally depending on the networks architecture. To be able to compare different security architectures, different proposed approaches need to be discussed. One problem in current research is the missing focus on different approaches for trust establishment in VANETs. Therefore, this paper surveys different security issues and solutions in VANETs and we furthermore categorize these solutions into three basic trust defining architectures: *centralized*, *decentralized* and *hybrid*. These categories represent how trust is build in a system, i.e., in a centralized, decentralized way or even by combining both opposing approaches to a hybrid solution, which aims to inherit the benefits of both worlds. This survey defines those categories and finds that hybrid approaches are underrepresented in current research efforts.**

Keywords–security; security issues; security architectures; VANET; MANET; ITS.

I. INTRODUCTION

This paper surveys different security architecture techniques for VANETs used in ITSs. Security plays a significant role in modern Cyber-Physical Systems (CPSs), which include intelligent transport systems. Trust establishment describes how trust is formed, which in return defines the fundamental security architecture of a system. The future of ITS is a networked one where vehicles and infrastructure do communicate to make traffic more efficient and safer. As vehicles are inherently mobile and self containing, the only way to communicate on the move is via wireless technology. Wireless has proven over time that getting security right in wireless technology is hard. An example of security done wrong is the utterly broken WEP technology, which is an acronym for "Wire Equivalent Privacy", but it was never able to fulfill that promise. Not only do security issues in regards to authentication exist but furthermore there are also security issues, like denial of service, replay or spoofing attacks, which vary in severity and easy of exploitability. But overall security in wireless technologies is boiling down to how trust is established and which methods and algorithms are used to secure the trust establishment.

One centralized approach for conveying trust is the mode of operation that is used in telecommunication standards all over the world, i.e., Global System for Mobile Communications (GSM), Code division multiple access (CDMA), Universal Mobile Telecommunications System (UMTS) or Long-Term Evolution (LTE). That mode of operation is building trust based on a shared symmetric secret. Until recently the weaknesses were discussed, like the important dependence of secrecy of this shared secret, but arguments were often discarded because of the needed effort to steal the secret key of each customer and the high security approach network operators supposedly are taking to secure those secrets. The treasure trove of leaked information by former National Security Agency (NSA) contractor Edward Snowden, showed the security community again once more how bad our assumptions were in this regard. As documents provided by the online publication THE INTERCEPT [1] show that American (NSA) and British spy agencies Government Communications Headquarters (GCHQ) managed to steal those important secrets directly from the manufacturer, in this case Gemalto. The theft means, on a technical level, that authenticity and confidentiality of a communication, supposedly secured by those stolen secrets, is compromised. The methods supposedly used to execute that theft raise many questions but this discussion might fit better in a legal or social publication and should not be discussed in this paper. Another rather recently uncovered attack on several banks (ca. 100 banks) around the world (ca. 30 countries) where a so called "Carnbanak cybergang" have stolen an estimated amount of 1 billion United States dollar (USD). While the used malware does not appear to be of very high sophistication, only the weakest trust relationship in computing, between a human and a machine was exploited via spear phishing, the orchestration, endurance and the targeted approach of the attackers where extremely remarkable, as reported by Kapersky Labs [2].

The underestimation, to what length state actors would go to achieve informational advantage in combination with how persistent and patient criminal actors are becoming, previously only attributed to state actors, goes to show how wrong and weak our current assumptions on cybersecurity have been. All central approaches bear the risk of being exploited by targeted attacks on the central trust anchors. And this is why we urge every researcher to reevaluate their assumptions and seek for alternative designs. Both currently developed major standards, IEEE WAVE [3] or ETSI ITS-G5 [4], favor a centralized security architecture, with trust rooting in central authorities, which represent a high value target for attackers to exploit.

 ISBN: 978-1-61208-427-5

Decentralized approaches could be such an alternative, by making such attacks much more risky and costly due to the distribution of trust relations. Distributed trust relations have their own issues, like performance or new attack opportunities not present in centralized architectures. Therefore, the combination of both architectural approaches, in this paper called hybrid approaches, could pose a overall security improvement, especially in the current environment with increasing proliferation of attack and exploitation techniques accessible to criminals and state actors alike. Attacking seams to be easier than defending, this is why we argue a easy to defend security architecture is paramount for any information system or network nowadays, especially in the field of ITSs, which are in focus of this paper.

As stated previously trust establishment can be achieved via a centralized way e.g., a Public Key Infrastructure (PKI), decentralized e.g., a Web of Trust (WoT) or by using a hybrid approach, which tries to combine the benefits of both approaches. Security issues in mobile ad-hoc networks are used to find solution for them and then categorizing those solutions into their general security architecture. We limit ourselves to some of the following major issues in ITS mentioned by various researches like Hubaux et al.[5], Lin et al.[6] or in an already summarized from by Yang [7].

- Impersonating by false, stole identities, Message spoofing or replay attacks
- Tampering with data in-transit
- Send false feedback to silence other vehicles
- Sinkhole attack via false routing information to effectively execute Man-in-the-Middle (MitM) attacks
- Sybil attack by creating virtual sock puppet identities to manipulate voting procedures to the attackers benefit
- An eclipse attack is similar to an sybil attack it specifically tries to split a network by using means of a malicious group of nodes
- Manipulating the network topology and disturbing node by connecting far away segments via a hidden tunnel (wormhole attack)
- Privacy violation caused by continues communication
- Denial of Service (DoS) by jamming signals or overloading specific nodes

In [8], Agrawal et al. present a short overview of different security issues and solutions with their objectives and draw backs. Mishra et al. [9] display a wide array of research effort in regards to security issues and solutions, which they think are important. A detailed introduction into VANETs is given by Raya et al. [10] they furthermore expand on, security issues and solutions in VANETs. Zhang[11] categorizes trust management for VANETs in three models: *Entity-oriented Trust Model*, *Data-oriented Trust Model* and *Combined Trust Model*. We differentiate various approaches to security issues in VANETs into three categories: *Centralized*, *Decentralized* and *Hybrid* as we think these categories describe the way trust is build better.

We define those categories in detail in the following Section II. Thereafter in separate sections we describe eight different security issues, already defined by Yang [7]. Each of these section contains solutions to its security issues, which are categorized according to our definition into: *Centralized*, *Decentralized* and *Hybrid* solutions. A summary of this paper is provided in the last Section III.

II. ANALYSIS

One of the main issues in ad-hoc-communication is trust. It is a basic problem in security to establish so-called trust anchors. Several models for trust management exist. The surveyed approaches with their assigned categories are listed in the summary table I.

Centralized: A central trust model may for example be implemented by a PKI. A PKI consists of one or more Certificate Authorities (CAs) that issue certificates to the participants of the system. The issue of certification may be delegated to Sub-CAs, resulting in a hierarchy of CAs. A certificate of a participant is considered to be legitimate, if it is possible to find a certification path from the certificate to a known and trusted CA. Several (yet unknown) Sub-CAs may be on the certificate path. Per se, all legitimate certificates are considered trusted. Hence, all the known and trusted certificates represent trust anchors for one participant. Using a PKI simplifies trust establishment to secure setup of trust anchors on an instance of the system.

Decentralized: A decentralized trust model may for example be a so WoT. In a WoT scenario, each participant of the whole network is also a CA and may express trust in a certificate of another participant. Each participant keeps a list of other participants of the system that are trusted and another list of participants that are trusted to express trust in other participants. As with the PKI, establishing trust in an unknown participant requires to build a trust path between the unknown participant and oneself. However, as no hierarchy exists, finding such a path is a hard task. Another approach to trust establishment are reputation models. No certificates are issued but the behavior of participants is monitored and trust values are assigned based on different attributes like former or expected behavior. Participants may exchange trust values of each other.

Hybrid: A hybrid trust model is one that makes use for example of a distributed PKI, which assigns identities to participants, mainly for liability reasons. This trust path is only used in case of an accident or when certain conditions are met. But the operational trust between participants is realized via a reputation system and only if enough evidence of bad or malicious behavior was recorded, the PKI infrastructure would step in to permanently revoke or destroy the cryptographic material of the offending node. Hybrid solutions are trying to combine both central and decentral approaches, to get the benefits of both approaches, like somewhat independence of central infrastructure or better privacy features.

A. Impersonation

Defending against replay or whole message spoofing attacks is usually done at a protocol level. If used communication

TABLE I. ANALYSIS OVERVIEW

Security Issues	Centralized	Decentralized	Hybrid	Ref.
Impersonation	[12],[5],[13]	[14],[15]	[16]	II-A
Data Tampering	[17],[12],[10]	[14]	[18]	II-B
Routing Attacks	[19],[20],[21]	[22],[23]	[13]	II-C
Sybil Attacks	[24]	[25],[26]	[27]	II-D
Eclipse Attacks	[28]	[25]	-	II-E
Wormhole Attacks	-	[29],[19]	-	II-F
Denial of Service	[13]	[30]	-	II-G
Privacy Violation	[12],[31],[16],[28],[32]	[17]	[33],[34]	II-H

 ISBN: 978-1-61208-427-5

protocols do not defend against those attacks a communication system, regardless its architecture will be hard to secure. Communication systems usually use some kind of identities to distinguish between different participants. Those identities usually need to be protected against impersonation to sustain distinguishability.

Centralized: In case strong identities are needed like in a system utilizing identity based cryptography [35] Sun et al. [12] propose storing identities in a tamper proof hardware to prevent identity theft. So do Hubaux et al. [5] they store their form of identity, called electronic license plate, in an event date recorder (EDR), similar to a black box in an aircraft. The EDR in return itself should be "protected [...] physically"[5]. Similarly Raya et al. [13] are using a "trusted component" in their protocols to store and protect identity data against theft.

Decentralized: Every participant in a VANET should have its own model of its vicinity and validate every piece of data received, according to Golle et al. [14]. They authenticated communication via public/private key pairs but they are self generated by each node and should be refreshed constantly [14]. Additionally, they propose using "location-limited channels"[15] to distinguish nodes. As an example of a "location-limited channel"[15] is the use of infrared signaling given by Golle et al. [14].

Hybrid: In an approach called "Efficient Decentralized Revocation Protocol"[16] (EDR) Wasef et al. propose a way to revoke trust in identities based on "probabilistic random key distribution technique and a novel pairing-based threshold scheme"[16]. It uses PKI but the revocation process is decentralized and facilitated by voting.

B. Data Tampering

Depending on how nodes are communicating in a VANET, whether it is single hop or multi hop communication, different opportunities arise for data tampering or manipulation. If transmitted data is not integrity protected, any intermediate system or bystander could change the information for its own benefit.

Centralized: One of the more complete approaches was proposed by Li et al. [17]. They based their scheme also on identity based cryptography [35]. But they extended it with blind signatures and one-way hash chains to provide mutual authentication, confidentiality and integrity, while preserving privacy. This approach is similar to that of Sun et al. [12], which also is based on identity cryptography and aims to deliver on the same security requirements [10].

Decentralized: Using a reputation system in conjunction with collecting and querying for additional data, to verify and attest trustworthiness of information is proposed by Golle et al. [14]. Every node builds up his own model of the network around him and validates data against it.

Hybrid: In [18], Zhang et al. present a scheme called "RAISE" a Roadside Unit(RSU)-adied message authentication scheme, which uses keyed-hash message authentication where the secret key is known by the RSU, which in return can therefore attest that the message is authentic. The proposed scheme is compatible with traditional PKI-based systems, further more it makes use of PKI as a fallback mechanism.

C. Routing Attacks

To prevent congestion in ad-hoc wireless environments nodes are listening to its neighbors and if a neighbor is better suited to forward messages it stops rebroadcasting messages. If an attacker could convince a node that he is better positioned, the attacker can silence other nodes. Which would make them effectively disappear from the VANET, so called silencing attacks. Also a vehicular ad-hoc network where bandwidth is limited, and far reaching connections to central systems needed to be routed through long range wireless communication technology like LTE or UMTS. Those communication technologies are expensive to use compared to a node posing as a high speed uplink or gateway reachable via ad-hoc communication, called sinkholing attack. This enables MitM attacks, where a malicious gateway can intercept or even alter the sent and received messages.

Centralized: One of the first secure routing protocols for VANETs were proposed by Eichler et al. [19] called AODV-SEC based on "Ad-hoc On-demand Distance Vector" (AODV). Lu et al. designed the "social-based privacy-preserving packet forwarding" [20] (SPRING) to be resistant against black holing attacks by utilizing road side infrastructure. Relying on PKI for strong identities but giving incentives, based on game theory, to nodes taking part in a mobile ad-hoc network, was proposed by Zhong et al. [21]. Sprite, "a simple, cheat-proof, credit-based system for mobile ad-hoc networks" [21] also needs a central Credit Clearance Service (CCS) to function.

Decentralized: In [22], Huang et al. propose a cluster based intrusion detection system to detect various attacks, among them sinkholing or blackholing. Their approach is focused on detection of those attacks and mitigation is left for the network to handle. The CONFIDANT protocol by Buchegger et al. [23] consist out of four entities present in each node: Monitor, Reputation System, Path Manager and Trust Manager. The Trust Manager collects events via the Monitor and uses the Reputation System to evaluate the events and the result of the evaluation are used by the Path Manager to adjust the rounting, to mitigate attacks like sinkholing.

Hybrid: To detect and respectively mitigate misbehaving nodes Raya et al. [13] propose two methods "Misbehavior Detection System (MDS)" and "Local Eviction of Attackers by Voting Evaluators (LEAVE)". When detecting a misbehaving node, LEAVE is used to degrade the attackers trust until a central certificate authority revokes its certificates. LEAVE is resilient to interference as long as colluding attackers are a minority.

D. Sybil Attacks

When protocols with voting procedures are used or if some kind of collaboration between nodes for making collective group decisions is needed, then a so called sybil attack could be used to influence protocols or decisions. This is done by creating sock puppets that the attacker controls to act on behave of him. In an vehicular environment, if an attacker would like to push the envelope, he and his sock puppets could simulate braking or congestion, and then tricking the victims into believing him. Protocols like the previously mentioned LEAVE Protocol by Raya et al. [13] have a certain threshold to, which they are resilient against a sybil attack. The important factor is the size of the sock puppet group in comparison to the amount of honest nodes.

Centralized: An easy protection against sybil attacks is the use of centrally enforced and distributed strong identities. Identities are created by a central authority and handed down to the nodes prior to their deployment as stated by Piro et

 ISBN: 978-1-61208-427-5

al. [24]. This process could be upfront or part of a VANET joining protocol. Either way a central entity knows to whom it has handed a specific identity. With autonomous vehicles at the horizon it may be even more compelling or tempting to use the vehicle identification number as such an id. This approach has many privacy implications, like unique traceable identities, or the central data storage would be a high value target for theft or intrusion.

Decentralized: In [25], Xiao et al. draft a technique called "Basic Signal-Strength-Based Position Verification" [25], which is used to verify the position by a claimer based on the signal strength. This technique is then used after collecting beacon messages to decide based on probability if there is a sybil node nearby and if so a statistic model is used to attribute the sybil nodes to one originating vehicle. Park et al. are using a timestamp based approach to detect sybil attackers [26].

Hybrid: An approach using timestamp series and RSUs issuing certificates was proposed by Park et al. [27]. The RSUs themselves have public private key pairs and a certificate from a central certificate authority. All vehicles must have the public key of the certificate authority pre-installed. Additional vehicles generate their own pair of keys. Similar timestamps series are identified as a sybil attack. To protect against sybil attack each data message must contain current timestamp certificate, RSU certificate, signed data and of course the data itself. If any inconsistencies occur the packets should be dropped. As the authors suggested by themselves [27] this approach is not suited for high traffic and urban scenarios, due to the spatial and temporal difference assumption falling apart.

E. Eclipse Attacks

An eclipse attack utilizes compromised neighbors to influence group decisions. It is also useful when the separation of nodes from other nodes weakens the whole network segment, by degrading the trust in the honest group while improving its own standing in the network. This approach usually eases and strengthens other attacks like DoS II-G.

Centralized: Quick and efficient removal of identified malicious nodes is key in protecting against eclipse attacks. Therefore Wasef et al. [28] proposed, based on a PKI system, not only a novel message authentication approach but also a quick certificate revocations approach to evict the trustworthiness of misbehaving nodes.

Decentralized: Some of the methods used to defend against sybil attacks also could be used to defend against eclipse attacks especially Xiao et al. [25] are trying to suppress sybil attacks in conjunction with opposite traffic flow and their ability to proof that they came from an upstream source.

Hybrid: **-** No hybrid approaches were found in literature.

F. Wormhole Attacks

When an attacker can control two nodes in different VANET segments and has a high speed link between those two, he can mount a so called wormhole attack. Illegal but correct traffic would originate from and to both ends of the tunnel, making vehicles suddenly appear in each others vicinity, while actually being in two remote locations. This type of attack could be the basis for executing other attacks, like sybil II-D, eclipse II-E or denial of service II-G attacks. A wormhole might be used by an attacker, to generate illegal traffic and let the nodes interfere with each trustworthiness in the connected segments, influence voting procedures or even cause a denial of service when the nodes in both segments revoke each others trustworthiness based on wrong positioning information.

Centralized: Assuming global network visibility is achieved, illegal traffic, which would be generated by a wormhole attack could be spotted by roadside units, acting as a sensor. The central network management system should then be able to correlate that the same traffic is visible in two remote locations. Mitigation of such an attack would only be a notice to affected nodes to discard traffic that is not in their vicinity. Most stronger responses like revoking the right to allocate a channel for communication would not harm the attacker in between but the nodes in their respective network segment. This could result in a DoS attack.

Decentralized: In [29], Safi et al. based their effort, like Eichler et al. [19], on the AODV Routing Protocol and enhanced it to include "geographical leashes" that should prevent the forwarding of packets from different geographic areas with additional packet authentication.

Hybrid: **-** No hybrid approaches were found in literature.

G. Denial of Service

Denial of service attacks are often used as distraction or an accompanying attack that should weaken the position of a system to ease the real attack or exploit. This type of issue is one of the harder ones to defend against. Because there are no purely technical means to defend against jamming attacks in wireless communication systems. Types of denial of service attacks include jamming of radio frequencies, traffic flooding or silver bullet attacks, where one specially crafted packet may be able to disrupt service.

Centralized: When a system needs a functioning PKI, like most of the mentioned approaches, or the one from Raya et al. [13]. A DoS attack could me mounted by creating a lot of identities and then report those same identities as malicious or fraudulent. This could result in a flood of certificate revocations, which could lead to DoS when revocation lists get to big or the revocation operation is computational intensive. To mitigate this threat Raya et al. [13] suggested the use of "Compressed Certificate Revocation Lists (RC^2RL)" and "Revocation of the Trusted Component (RTC)" protocols.

Decentralized: For VANETs Hamieh et al. [30] described a method to detect on going jamming attacks. They focused on attacks where the jammer is only sending when his hardware is allowed to, he abides the rules of the underlying IEEE 802.11p Standard. Their model is based on time correlation of errors and correct receptions to detect the presence of a jamming attack.

Hybrid: **-** No hybrid approaches were found in literature.

H. Privacy Violation

In a cooperative system where every neighboring node should have all the needed information to make intelligent decisions on its own and for the group, it is clear that all this information needs to be communicated. Therefore, when every node broadcasts his position, trajectory, acceleration, route or other data, basically a profile of the driver could be created. If this data is readable by everybody in the vicinity, somebody just needs to set up an antenna and can now make statistics where and when people are driving, when traveling past him.

Centralized: To protect privacy and making tracking harder most approaches use pseudonyms and rotating them, like [12], [31]. But everybody does it slightly different, Sun

 ISBN: 978-1-61208-427-5

et al. [12] are using "preloading [...] pseudonym(s)" whereas Choi et al. [31] use generation of public keys by deriving it from the secret id only known to an authority and the vehicle itself. While still allowing the verification and certification based on time stamps and other public key parameters. But almost all approaches [16], [28], [32], found during our survey are using PKI to guarantee authenticity and non-repudiation. The latter one supposedly for liability reasons.

Decentralized: To preserve privacy Li et al. [17] presented a scheme called "SECSPP" utilizing non interactive identity based cryptography and a blind signature scheme for allowing anonymous usage of RSU services. Anonymous confidential communication between the RSU and vehicles make tracking or eavesdropping harder and more expensive.

Hybrid: A cluster based architecture utilizing PKI, theshold cryptography and location limited side channel [15], like license plate recognition is proposed by Bechler et al. [33] to secure ad-hoc communication, similar to an approach by Zhou et al. [34]. To adapt to different security levels the approach by Bechler et al. [33] supports 4 different modes of operation: no encryption, cluster key encryption, public key directly exchanged and public key certified by a distributed certificate authority, in this case the cluster heads.

III. Conclusion

Some security issues in ITSs are hard or outright impossible to mitigate on a purely technical basis, this is why we did not consider them in our review. Examples for this type of attacks are, jamming or physical tampering. An attacker with a radio frequency jammer can suppress any meaningful communication [5]. Often the solution to physical tampering is to even better tamper proof those devices, like sensors or Electronic Control Units (ECUs). This climaxes often in the inclusion of a Trusted Platform Module (TPM), which shifts the responsibility and trust to the manufactures of those components. But as explained in the introduction I, trust in those supposedly highly secure entities has been shattered in the recent years. Therefore relying on them can be the Achilles heal of a system. Besides those doubts there are many solutions for centralized architectures and some decentralized ones. We were able to find suitable hybrid solutions in the literature for only five out of eight security issues. Often only one hybrid solution could be found for a specific security issue. Our findings are summarized in the Table I. We therefore conclude that hybrid approaches are underrepresented in current research, which might be an indicator that further research is needed or that hybrid approaches appear to be fruitless endeavors. To answer those questions further research, including a comparative study, needs to be conducted. The direction the standardization efforts, by IEEE and ETSI, are heading, is towards centralized architectures with all benefits and weaknesses. Those will set the mark against all other solutions have to prove themselves. Eventually decentralized solutions could be considered for integration in those standards if proven beneficial.

References

[1] J. Scahill and J. Begley. The Great SIM Heist: How Spies Stole the Keys to the Encryption Castle. [Online]. Available: https://firstlook.org/theintercept/2015/02/19/great-sim-heist/ [retrieved: jul, 2015]

[2] Kaspersky Lab HQ. Carbanak_apt_eng.pdf. [Online]. Available: http://25zbkz3k00wn2tp5092n6di7b5k.wpengine.netdna-cdn.com/files/2015/02/Carbanak_APT_eng.pdf [retrieved: jul, 2015]

[3] Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society, "1609.0-2013 - IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture," IEEE Std 1609.0-2013, Mar. 2014, pp. 1–78, bibtex: 6755433. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6755431

[4] ETSI TR 102 962, "ETSI TR 102 962 v1.1.1 (2012-02) intelligent transport systems (ITS); framework for public mobile networks in cooperative ITS (c-ITS)," feb 2012, pp. 1–63.

[5] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," IEEE Security & Privacy Magazine, vol. 2, no. LCA-ARTICLE-2004-007, 2004, pp. 49–55.

[6] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Security in vehicular ad hoc networks," Communications Magazine, IEEE, vol. 46, no. 4, 2008, pp. 88–95.

[7] W. Yang, "Security in vehicular ad hoc networks (vanets)," in Wireless Network Security. Springer Berlin Heidelberg, 2013, pp. 95–128. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36511-9_6

[8] A. Agrawal, A. Garg, N. Chaudhiri, S. Gupta, D. Pandey, and T. Roy, "Security on vehicular ad hoc networks (vanet): A review paper," International Journal of Emerging Technology and Advanced Engineering, vol. 3, 2013, pp. 231–235.

[9] B. Mishra, P. Nayak, S. Behera, and D. Jena, "Security in vehicular adhoc networks: a survey," in Proceedings of the 2011 International Conference on Communication, Computing & Security. ACM, 2011, pp. 590–595.

[10] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," Journal of Computer Security, vol. 15, no. 1, 2007, pp. 39–68.

[11] J. Zhang, "A survey on trust management for vanets," in Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on. IEEE, 2011, pp. 105–112.

[12] J. Sun, C. Zhang, and Y. Fang, "An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks," in Military Communications Conference, 2007. MILCOM 2007. IEEE. IEEE, 2007, pp. 1–7.

[13] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," Selected Areas in Communications, IEEE Journal on, vol. 25, no. 8, 2007, pp. 1557–1568.

[14] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," in Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks. ACM, 2004, pp. 29–37.

[15] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks." in NDSS, 2002.

[16] A. Wasef and X. Shen, "Edr: Efficient decentralized revocation protocol for vehicular ad hoc networks," Vehicular Technology, IEEE Transactions on, vol. 58, no. 9, 2009, pp. 5214–5224.

[17] C.-T. Li, M.-S. Hwang, and Y.-P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," Computer Communications, vol. 31, no. 12, 2008, pp. 2803–2814.

[18] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "Raise: an efficient rsu-aided message authentication scheme in vehicular communication networks," in Communications, 2008. ICC'08. IEEE International Conference on. IEEE, 2008, pp. 1451–1457.

[19] S. Eichler, F. Dotzer, C. Schwingenschlogl, F. J. F. Caro, and J. Eberspaher, "Secure routing in a vehicular ad hoc network," in Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, vol. 5. IEEE, 2004, pp. 3339–3343.

[20] R. Lu, X. Lin, and X. Shen, "Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[21] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol. 3. IEEE, 2003, pp. 1987–1997.

[22] Y.-a. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks. ACM, 2003, pp. 135–147.

[23] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confi-

 ISBN: 978-1-61208-427-5

dant protocol," in Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing. ACM, 2002, pp. 226–236.

[24] C. Piro, C. Shields, and B. N. Levine, "Detecting the sybil attack in mobile ad hoc networks," in Securecomm and Workshops, 2006. IEEE, 2006, pp. 1–11.

[25] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in vanets," in Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks. ACM, 2006, pp. 1–8.

[26] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," vol. 51, no. 12, 2007, pp. 3448–3470. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S138912860700062X

[27] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against sybil attack in vehicular ad hoc network based on roadside unit support," in Military Communications Conference, 2009. MILCOM 2009. IEEE. IEEE, 2009, pp. 1–7.

[28] A. Wasef and X. Shen, "Maac: Message authentication acceleration protocol for vehicular ad hoc networks," in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE. IEEE, 2009, pp. 1–6.

[29] S. M. Safi, A. Movaghar, and M. Mohammadizadeh, "A novel approach for avoiding wormhole attacks in vanet," in Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on. IEEE, 2009, pp. 1–6.

[30] A. Hamieh, J. Ben-Othman, and L. Mokdad, "Detection of radio interference attacks in vanet," in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE. IEEE, 2009, pp. 1–5.

[31] J. Choi and S. Jung, "A security framework with strong non-repudiation and privacy in vanets," in Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE. IEEE, 2009, pp. 1–5.

[32] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]," Wireless Communications, IEEE, vol. 17, no. 5, 2010, pp. 22–28.

[33] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," in INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4. IEEE, 2004, pp. 2393–2403.

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

[35] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in CryptologyCRYPTO 2001. Springer, 2001, pp. 213–229.

 ISBN: 978-1-61208-427-5

A Novel Financial Instrument to Incentivize Investments in Information Security Controls and Mitigate Residual Risk

Pankaj Pandey
Gjovik University College, Norway
University of Antwerp, Belgium
Email: pankaj.pandey2@hig.no; pankaj.pandey@uantwerpen.be

Steven De Haes
Antwerp Management School, Belgium
University of Antwerp, Belgium
Email: steven.dehaes@uantwerpen.be

***Abstract*—Recent cyber-attacks on various organizations indicate that even the most sophisticated technical controls are vulnerable. Furthermore, due to the problem of misaligned incentives it is inevitable to achieve absolute protection with technical controls against the risks and its impact. Thus, there is a space for alternative risk management methods. However, there is a lack of an (effective) financial mechanism to incentivize coordinated efforts by stakeholders in addressing the problem of information asymmetry, negative externality, and free-riding in the information security ecosystem. Therefore, we propose a novel financial instrument called information security financial instrument to incentivize investments in collaborative and multistakeholder initiatives to develop and implement stronger defense systems. The mechanism can contribute to an improvement in information security environment in a time bound manner. We have used a case-study to demonstrate the application of the information security financial instrument. Furthermore, we have analyzed the information security financial instrument against a set of requirements and its usefulness over cyber-insurance in incentivizing investments in information security mechanisms to manage risks. In our analysis, we found that information security financial instruments can be a solution to address (at least to some extent) various economic problems in the information security domain.**

Keywords–Information Security; Security Economics; Risk Management; Financial Instrument.

I. INTRODUCTION

In today's technology-driven world, where organizations are heavily dependent upon information and communications technology, any attack on the technology infrastructure, and services offered over a computer network may lead to operational disruptions. The information and communication infrastructure (cyber ecosystem) face a wide variety of risks posed by a variety of threats such as distributed denial of services (DDoS) attacks, intrusion, eavesdropping, etc. These risks if materialized may have a huge negative impact on the organization including a negative impact on profits, brand value, and reputation. Furthermore, a successful cyber attack on the company may lead to negative impact on stock prices and overall corporate value [1]–[3]. Therefore, to reduce the likelihood and impact of the information (cyber) security risks, organizations have traditionally resorted to technical controls such as antivirus software, firewall, intrusion detection systems, intrusion prevention systems, and so on. However, cyber-attacks on various organizations such as JP Morgan [4], SONY [5], Target [6], and many more [4] indicate that even the most sophisticated technical controls are vulnerable.

When pursuing information security from an economic perspective, the failure of technical controls in providing 100% defense against the information security threats can be explained with the following reasons: (i) The problem of 'lemons market' [7], i.e., security product vendors do not have enough incentives to ship robust products in the market; (ii) The problem of misaligned incentives [8], i.e., information security stakeholders such as users (individual or organizations), security product vendors (e.g., McAfee, Symantec), cyber-insurance providers (e.g., Zurich Insurance) and regulatory bodies (e.g., financial markets regulator SEC in USA, Insurance regulator, regulatory bodies dealing with data protection and privacy, etc.) have misaligned incentives; (iii) The problem of 'tragedy of commons' [9], i.e., the issue of negative externalities and free riding in the network. In the light of the barriers mentioned above, it is inevitable to achieve near 100% protection against the risks and its impact, thus creating a space for *alternative risk management methods*.

Problem Statement: Lack of an (effective) financial mechanism to incentivize coordinated efforts of stakeholders in addressing the problem of information asymmetry, misaligned incentives, negative externality and free-riding in information (cyber) security ecosystem.

Motivation: Currently, cyber-insurance is only commercially available financial product that can be used to mitigate residual information security risks [10]. The proponents of cyber-insurance argue that it has the potential to align the incentives of security product vendors, users, and cyber-insurance providers, thereby creating a robust information security environment. However, there is a very little evidence to suggest that the cyber-insurance products can improve the network security by providing adequate incentives to organizations and individuals to invest aptly in information security controls [11][12].

Some researchers have mathematically proved that the cyber-insurance markets are inefficient [13][14]. These researchers have reported that though the cyber-insurance products satisfy all the other stakeholders but they fail to satisfy the regulatory bodies and sometimes the cyber-insurer provider itself. The regulatory bodies are unsatisfied due to the sub-optimal network robustness occurring due to under-investment in security controls by the network users. On the other hand, due to the interdependent and correlated nature of information security risks the uncertainty about the quantum of risk

 ISBN: 978-1-61208-427-5

exposure leads to the fear of systemic and huge losses for cyber-insurance providers. The notion of making no profits (or facing huge losses) in the future leads to dissatisfaction in cyber-insurance providers.

Thus, in absence of adequate market mechanisms for risk acceptance, the interest of entities who wish to transfer their risks and those who are willing to accept the risk by means of pooling and necessary expertise, are reduced [15].

Objective: To develop a financial instrument to address the problem of misaligned incentives and incentivize the stakeholders in making coordinated efforts in improving the information security ecosystem.

Contributions:

1) Developed a novel financial instrument called Information Security Financial Instrument (ISFI) to incentivize coordinated efforts of information security stakeholders (investors) in improving the information security ecosystem in a time bound manner.
2) Demonstrated the application of the financial instrument to improve the performance of a specific firewall.
3) Analyzed and explained the usefulness of the information security financial instruments in dealing with the problem of information asymmetry, negative externality and free riding in the information security domain. Furthermore, we analyzed the usefulness of the instrument as a risk management tool.
4) Contributed to the knowledge base of interdisciplinary research on information security economics.

The remainder of the paper is structured as follows: Section 2 presents an overview of the research method followed for the article. Section 3 presents an overview of the background work. Section 4 identifies the requirements for information security financial instruments. Section 5 describes the proposed information security financial instrument. Section 6 demonstrates the application of the proposed information security financial instrument. Section 7 presents an evaluation of the information security financial instruments. Section 8 concludes the article with conclusion and directions for future research.

II. Research Method

The research follows the Design Science Research Approach (DSRA). DSRA is useful when innovations and ideas are created for the development of technical capabilities and products that will be instrumental in effective and efficient process development for artifacts [16]. A process flow model for DSRA is shown in Figure 1.

A. Explicate Problem

The first step is to formulate the initial problem, justify its importance and investigate the underlying causes [16].

To explicate the problem we started with examining the literature on information security investment models, and currently available market methods and financial instruments for the management of information security risks. This enabled us in identifying the gaps in existing methods of (financial) risk management in information security domain. The identified problem is given as the problem statement in Section 1, and we have explained the background issues in Section 3.

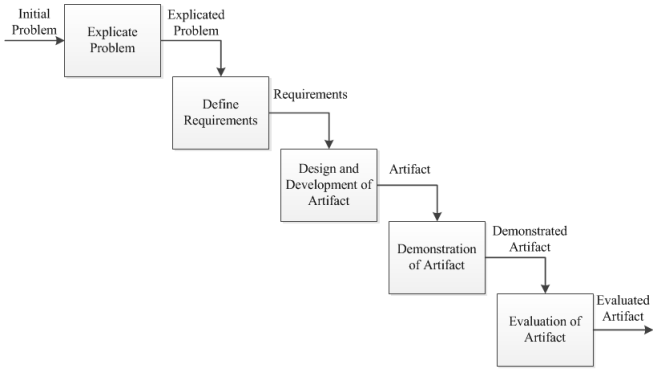


Figure 1. Process Flow Model for Design Science Research Approach [16].

B. Define Requirements

The second step is to identify and outline an artifact to address the explicated problem and to elicit requirements for the artifact [16]. A requirement is the property of the artifact that is desired by stakeholders in practice and is used for design and development of the artifact. A requirement can be functional, structural, or environmental in nature. The requirements for the artifact to address the problems identified in the previous step are given in Section 4.

C. Design and Development of the Artifact

The third step leads to the creation of an artifact that fulfills the requirements identified in the previous (second) step. This includes designing the functionality and structure of the artifact [16]. The functionality and structure of the artifact are explained in Section 5.

D. Demonstration of the Artifact

The fourth step proves the feasibility of the artifact by demonstrating its use in one case. Primarily, it consists of descriptive knowledge explaining the working of the artifact in one situation [16]. The demonstration shows that the artifact can, in fact, solve the problem (or some aspects of it) in the illustrative case. This demonstration can be considered as a weak form of evaluation. It indicates that if the artifact can address the problem in one situation; then it might be able to address the problem in other situations as well [16]. We have demonstrated the use of the artifact in Section 6.

E. Evaluation of the Artifact

The fifth step is to evaluate the artifact. This determines the extent to which the artifact can solve the explicated problem and its requirements [16]. An evaluation strategy can be an ex-ante or ex-post on the one hand and naturalistic or artificial on the other [16]. An ex-post evaluation implies that the artifact is evaluated without being fully developed or used. An ex-post evaluation implies that the artifact is evaluated after it has been implemented. A naturalistic evaluation implies that the artifact is evaluated in practice for which it is developed. An artificial evaluation implies that the artifact is evaluated in an artificial and contrived setting.

We have evaluated our artifact in Section 7 against the explicated problem and its requirements. We have used the 'informed argument' form of evaluation. Informed argument

 ISBN: 978-1-61208-427-5

form of evaluation is an ex-ante, artificial evaluation method, and it consists of arguments from the developers of the artifact [16]. In this case, researchers evaluate the artifact by reasoning and arguments for its usefulness in meeting the defined requirements and solving the explicated problem. Informed argument form of evaluation is often used to evaluate the artifacts that are highly innovative and are still immature [16].

III. Background Work

This section looks at the information security market from an economic perspective. In an efficient market for information security, buyers and sellers, both are expected to have sufficient information about the products. However, this is not currently the case. The information security goods are traded in markets with insufficient information similar to "The Market for Lemons" and "Market for Insurance". The following subsections explain the problems of information asymmetry, externality, and free-riding with the well-established economic theories.

A. The Market for Lemons

If the buyers lack information about the good and it then suggests that the sellers have sufficient information, then there is an asymmetry of information between the buyers and the sellers. This can be explained with the theory of "The Market for Lemons" proposed by George A. Akerlof [7]. Akerlof introduced "The Market for Lemons" with the question of why there is a "large price difference between new cars and those which have just left the showroom" [7]. He analyzed the rules of a market with information asymmetry between the buyer and the seller. He argued that a typical buyer of a used car cannot distinguish between the good cars and the bad cars (termed as "lemons"), as unlike the seller, the buyer does not know the true history of the used car. In such a scenario, the buyer is suspicious about the condition of the (good) car and is thus unwilling to pay more than the price of lemon (bad car). This type of market condition leads to under-supply of good condition used cars.

"The Market for Lemons" when mapped on to information security suggests that security product vendors do not have sufficient incentive to provide adequate security. It suggests that information security is a trust good and is not visible to the buyer. As a buyer cannot differentiate between the secure and insecure products, the product is traded at the price of insecure products (lemons). This leaves little incentive for the security product vendor to invest in the development of secure products. The security product vendor would rather prefer to have a less secure product and reach the market first to capture the market share or to invest in features that are more visible to the buyer.

B. Market for Insurance

Logically, it is unacceptable to suggest that only buyers lack the information and sellers have that information. Rothschild and Stiglitz examined the market of insurance as one "in which the characteristics of the commodities exchanged are not fully known to at least one of the parties" [17]. They claimed that "not only may a competitive equilibrium not exist, but when equilibria exist, they may have strange properties. In the insurance market, sales offers do not specify a price at which customers can buy all the insurance that they want, but instead consist of a price and a quantity – a particular amount of insurance that the individual can buy at that price. Furthermore, if individuals were willing or able to reveal their information, everybody could be made better off. By their very being, high-risk individuals cause an externality: the low-risk individuals are worse off than they would be in the absence of the high-risk individuals" [17]. This has an echo of "The Market for Lemons" but it is like a counter theory (mirror image) to Akerlof's work. Rothschild & Stiglitz assumed that "individuals know their accident probabilities, while companies do not" [17]. This is information asymmetry.

As discussed, the problem of information asymmetry has a negative effect on the insurance ecosystem, where it is difficult to distinguish between the high-risk and low-risk user types. This is commonly known as the problem of adverse selection. Similarly, users purchasing insurance policies when they know that they are highly likely to get affected, and they adversely affect the loss probabilities of the insurance providers. This is termed as the problem of moral hazard.

The theory of 'Market for Insurance' when mapped on to information security suggests that security product vendors know (at least to some extent) about the vulnerabilities in their products, however the users of the product are unaware about the vulnerabilities. Similarly, individuals and organizations purchasing cyber-insurance products have some information about the weaknesses in their defense system. However cyber-insurance providers lack a standard and evidence-based tool to check the strengths and weaknesses of the system. This information asymmetry leads to higher premiums, a large number of exclusions and the liability issue.

C. The Tragedy of Commons

The infrastructure of information and communication technology is largely interconnected and thus poses a challenge of collective security efforts by the participants. In economic terms, this can be explained with the theory of "Tragedy of Commons" proposed by Garrett Hardin [9]. According to the theory of the tragedy of commons, individuals acting rationally and independently in their self-interest with no consideration to long-term best interests of other members of the group would eventually lead to depletion of the common resources.

"The Tragedy of Common" when applied to information security domain explains the unwillingness of users to demand high security products. In a large distributed network, risks (and benefits) are spread over a set of nodes and are correlated. Thus, the information security is the property of the network and is not limited to its individual nodes. An investment in information security controls by one user to counter its risk exposure strengthens its security, and the node will strongly defend against the attacks. Thus, the benefit of defense gets propagated to other nodes in the network, and this is called 'positive externality'. Similarly, if the network is attacked, say by botnets, and one of the nodes gets corrupted then the risk of attack is propagated to other nodes leading to higher expected loss. In such a scenario, the cost (impact) of the attack is distributed between all the nodes. This is called 'negative externality'.

This suggests that the risks and benefits are all distributed between the nodes. This leads to a situation where individual nodes do not have a strong incentive to invest in information security unilaterally. They all tend to take a 'free-ride' on the

 ISBN: 978-1-61208-427-5

investments made by other nodes, thus depleting the common resource (security).

IV. Requirements for Information Security Financial Instruments

A report from World Economic Forum states that "No one organization can resolve the *(cyber-security)* issue by itself and a collaborative, multistakeholder approach must be taken; even competitors in a given industry must become partners in the effort to ensure a stable and trusted environment" [18]. Another report from World Economic Forum states that "Opportunities will emerge for new businesses in insurance or risk markets to help businesses mitigate the potential downside from cyber risks" [19].

As shown in Figure 2, Risk markets are one of the two ways to deal with systemic risk in information security domain [18]. Risk markets can provide a variety of financial instruments such as indemnification, insurance and structured risk-transfer solutions for an organization to address the information security risks [19].

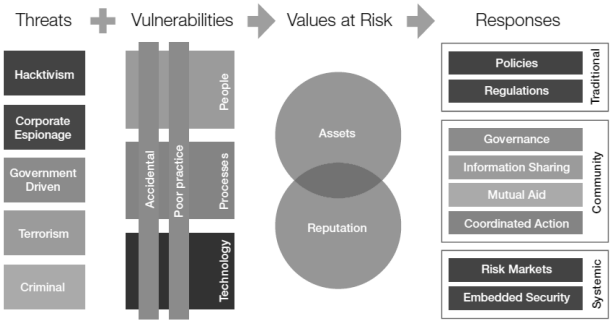


Figure 2. Cyber Risk Framework [18].

Therefore, keeping in view the problems identified in Section 3 of this article and the findings of World Economic Forum in [18][19], we have identified the following requirements for the Information Security Financial Instruments(ISFI):

Functional Requirements

- ISFI should incentivize coordinated efforts and investments in strengthening the security ecosystem.
- ISFI should tie the financial returns to the achievement of measurable or observable impact (performance or results), as specified in contract's specification.
- ISFI should fix accountability on the information security stakeholders, such as on project executors, or information security product vendors, to achieve the desired objectives.
- ISFI should clearly define the return structure.
- ISFI can be designed as an equity, debt or convertible instrument.

Usability Requirements

- Only verified traders/investors should be allowed to deal with ISFI.
- ISFI should be traded in a transparent environment.
- ISFI should be listed at (traded via) a regulated platform.
- ISFI should allow anonymous trading/investing.
- ISFI should be traded in a manipulation resistant environment.
- ISFI should be traded at low transaction cost.
- ISFI should be traded in a liquid environment.

V. Information Security Financial Instrument

This section presents a novel financial instrument, called Information Security Financial Instrument (ISFI) to incentivize investments in collaborative and multistakeholder initiatives to develop and implement stronger defense systems. The returns on ISFI are linked to the achievement of certain security objectives and mitigation of underlying risks. An application scenario for ISFI is shown in Figure 3.

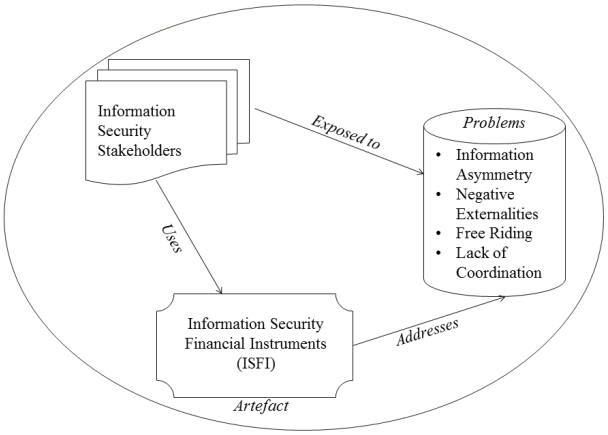


Figure 3. An Application Scenario for ISFI.

The ISFI can be implemented in at least following two forms:

- **Results based Information Security Financial Instruments:** This type of instruments provides a mechanism and strategies designed to tie purposefully the investments in information security controls and risk mitigation methods and thus incentivize the efficient allocation of resources provided by security stakeholders. Properly designed and well-implemented results based information security financial instruments may result in improvement in quality and timely delivery of (more) secure products, lower risk exposure, a shift towards a result oriented rather than a ship next day approach, and an improved security ecosystem. However, these benefits come at the expense of some opportunity costs, the need to monitor and test the performance, and exposure to the risk of incorrect incentive system.
- **Information Security Performance Instruments:** Information security performance instruments, if designed properly, can be helpful in achieving the long-term improvement in the information security ecosystem, increasing efficiency and creating a favorable environment to attract investment capital. Information security performance instruments can be designed to meet security goals for critical public infrastructure

 ISBN: 978-1-61208-427-5

such as power grids, in the oil and gas sector, financial sector and others, and include time-bound performance goals against which the performance of the service operator is measured. ISFI can be useful in sourcing funds for the projects where traditional funding sources are not (or less) useful.

The process of designing an information security financial instrument is shown in Figure 4.

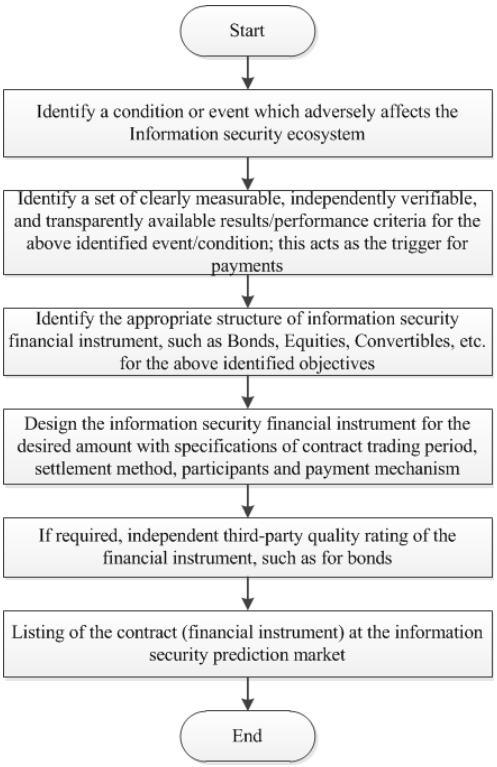


Figure 4. Process of Designing and Trading an ISFI.

The six processes shown in Figure 4 are explained in the following subsections.

A. Identification of Security Objectives

The design of useful information security financial instruments to incentivize investments in improving cyber-defense systems and risk mitigation depends upon the identification of useful 'objectives'. The incentive system depends upon the achievement of desired and predefined underlying security objectives to which returns are linked.

The underlying objectives of an information security financial instrument can be structured in at least the following five ways:

- *Reductions*: For example, reduction in number of vulnerabilities in a piece of software.
- *Improvements*: For example, improvement in the cyber-intelligence tool used by a law enforcement agency.
- *Increasing*: For example, increase in the detection of new viruses in the cyber space.
- *Decreasing*: For example, decrease in the false acceptance rate of a biometric authentication system.
- *Compliance Goals*: For example, meeting the industry compliance and regulations, such as HIPAA, SOX, etc., and thus avoiding any fine for violation of the norms.

Table 1 presents a set of entities (stakeholders) that can play a vital role in identification of the underlying objectives for information security financial instruments.

TABLE I. TYPICAL STAKEHOLDERS OF ISFI.

Issuer	Issuer entity can be a government body, regulatory body or a financial institution interested in achieving an information security objective. Also, an industry body charged with the responsibility of achieving specific goals or a beneficiary of achievement of objectives can be an issuer. Cyber-Insurance providers, Reinsurance providers, security product vendors, etc., can be instrumental in identifying the information security objectives for which financial instruments are to be issued.
Investors	Investors are interested in allocating capital and resources to fund large scale information security projects (critical infrastructure like power grids, implementation of privacy policy at a country/industry/EU level, etc.), earn profits tied to the achievement of objectives, and hedge the risks associated with the underlying objectives. Insurance providers, reinsurance providers, project executors, users, etc., can be investors.
Executors	They are the entities with the responsibility of achieving the objectives as specified in the contract (financial instrument) description. Depending upon the underlying objectives, executors can be software product vendors, vendor's competitors, security researchers, etc.
Clearing House	Clearing house acts as an inter-mediator between the issuer, investors and executors. Clearing house manages the credit risk, trader/investor verification, acts as an absolute authority on settlement of contracts, and an independent third party for the verification of claims.

Furthermore, it is not necessary to have clear demarcation of roles between the above entities and multiple functions can be performed by a single or a combination of above entities.

B. Identification of Payment Trigger Criteria

The payment trigger criteria should be clearly defined, be measurable or impact observable, transparent, and verifiable by an independent third party. Further, the contract specification on the trigger criteria should avoid any present and foreseeable conflict of interests between the issuing entity and other administrative stakeholders.

The ISFIs can be structured with various types of measurable and observable impact criteria, such as 'reduction' (e.g., 2%, 5%, etc.) in number of vulnerabilities discovered in a piece of software (thereby making the software more secure), an 'increase' (e.g., 3%, 7%, etc.) in accuracy of a new biometric based authentication system, and so on.

Table 2 presents a set of payment trigger criteria for ISFIs (equities, bonds, and convertibles).

C. Types of Financial Instruments

The ISFI can be designed as 'debt', 'equity', or 'convertible' instruments.

- *Debt Instrument:* A debt instrument is a contract between a lender and a borrower under which borrower borrows money in exchange of payments of the principal amount and fixed interests over a defined period. One such instrument is a 'bond'.
  The issuer i.e., the indebted entity issues a bond specifying the coupon i.e., interest rate that will be paid with the principal amount on the maturity date. The

TABLE II. PAYMENT TRIGGER CRITERIA FOR ISFI.

Trigger Criteria	Examples
Performance Index	Such as ISE Cyber Security ETF (HACK) [20], UK Cyber Vulnerability Index [21], Global Cybersecurity Index (GCI) [22], Index of Cyber Security [23]
Results Indicators	Such as Technological Indicators (improvement in performance of antivirus software,firewall, etc.), Process and Procedural Indicators (compliance with data management & privacy policies, such as deletion of user data after certain period, etc.), so on.
Customized Indicators	Such as a combination of performance index and result indicators, Qualitative analysis of security strength, penetration of antivirus, firewall and other security defense systems, so on.

two key features that determine the interest rate on the bond are duration and credit quality. An independent third party can be involved to certify the credit quality of the bond.

- *Equity Instrument:* Equity instruments are tradable assets i.e., tradable capital packages with unique structures and characteristics. Equity instruments are different from debt instruments in a way that they provide some control and ownership of the business. A commonly known equity instruments is stock.
- *Convertible Instrument:* Financial instruments that can be converted to common stocks are known as convertibles, such as bonds and preferred shares. For example, holders of convertible bonds are allowed to convert their position to equities at an agreed price.
  Convertible financial instruments are attractive to investors looking for higher returns than bonds and equities. For example, convertible bonds will have lower coupon rate than traditional bonds. However, the option of converting the bond to common stocks provides an added value to the holder.

Table 3 presents three types of ISFIs and corresponding trigger criteria for payments.

TABLE III. TYPES OF ISFI AND PAYMENT TRIGGER CRITERIA.

ISFI Type	Payment Trigger Criteria
Information Security Equity	Result Indicators
Information Security Bonds	Performance Indices
Information Security Convertibles	Pre-specified Indicators

D. Contract Specifications

Next step in the process of creation and trading of ISFIs is the specification of contracts. We have identified a set of specifications which needs to be considered when creating an ISFI. Table 4 presents a template with the specifications identified for the information security contracts.

E. Return Structure

The returns on ISFIs can be structured in a variety of ways depending on the objectives of the issuing entity. The triggers for returns are linked to the achieving the specific objectives as specified in the contract description. Table 5 presents a set of return structures for ISFIs.

The ISFIs can be designed with other types of return structures, or a combination of return structures can be used.

TABLE IV. TEMPLATE FOR SPECIFICATIONS OF ISFI.

Issuer	
Objective of the Funding	
Benchmark Measurement Criteria	
Total Funding Required	Amount : Currency :
Project Start Date	
Project End Date	
Information Security Financial Instrument Type	
Transferable Instrument	
Decision Criteria	
Initial Benchmark Value	
Minimum Investment Required	Amount : Currency :
Eligible Investors	
Independent Third Party Quality Rating Required	Yes/No
Independent Third Party Verification Required	Yes/No
Management Fee	
Know Your Trader/Investor Required	Yes/No
Return Structure	
Pay-Off Horizon	
Bonus Payment	Yes/No
Trigger for Bonus Payment	If applicable

TABLE V. RETURN STRUCTURE FOR ISFI.

Fixed Return Structure	The returns i.e., bond yields or stock dividends are fixed and based on achievement of pre-determined objective. For instance, improvement in performance of 'XYZ' firewall in defending against 'UVW' types of attacks by 10% in 01 year will provide a return of 3%
Increasing Return Structure	In this case, returns are proportionately linked to increase in performance or quality or impact outcomes. For instance, for every 1% improvement in performance of 'XYZ' firewall in defending against 'UVW' type attack will provide 0.1% return
Tiered Return Structure	In this structure, returns depend upon the level of outcomes, i.e., the return structure is tiered (increase or decrease). For instance, a 5% improvement in performance of 'XYZ' firewall will yield 3% return, and an improvement of 10% will yield 7%, and so on.
Decreasing Return Structure	In this structure the return decreases with decrease in performance outcomes. This leads to reduction in interest disbursements and thus, creates a tangible reward for the issuing entity.

For instance, a fixed return structure can be used up to a certain level and then a tiered return structure is used, etc.

The structure of ISFIs will depend upon the specific information security objectives of issuing entity. For instance, for improvement in performance of a particular cyber-intelligence tool which is used by government organizations and has been developed by or in collaboration with a private organization, then a tiered return structure can be used for the ISFI. In this case, the instrument will have a base return and a bonus return will be awarded if the performance of the said cyber-intelligence tool is assessed to be above the threshold as specified in the contract (financial instrument) specifications. Table 6 presents a set of ISFIs, respective return structures and trigger criteria.

TABLE VI. ISFI, RETURN STRUCTURE, AND TRIGGER CRITERIA.

ISFI Type	Return Structure	Trigger Criteria
Bonds	Fixed Return	Performance Index
Equity	Increasing Return	Result Indicators
Convertibles	Decreasing Return	Customized Indicators
Convertibles	Tiered Return	Customized Indicators

 ISBN: 978-1-61208-427-5

F. Listing of Contracts

Once the ISFIs are created they can then be traded over-the-counter (OTC) or they can be listed at the information security prediction market to allow trading of the contracts. Information security prediction market is the preferred platform, as it is expected to facilitate information elicitation, trading transparency, lower transaction cost, liquidity, efficiency and manipulation resistance.

VI. Example Application

In this section, we demonstrate the application of ISFI in improving the performance of firewalls developed by Europe-based organizations against a particular 'UVW' type of attacks. As Firewalls are the first line of defense against information security attacks, an improvement in performance of firewalls is highly important in addressing the 'public goods' nature of information security and addressing the problem of negative externality and free riding. An application scenario of information security bonds in strengthening the security ecosystem is shown in Figure 5.

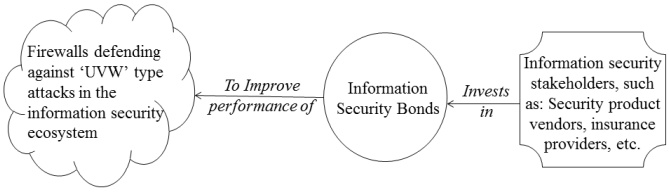


Figure 5. Application of ISFI in Strengthening Information Security Ecosystem.

An information security bond issued by the association of information security product vendors in Europe, to improve the performance of firewalls developed by Europe based organizations to defend against the 'UVW' type of attacks is shown in Table 7.

If an investor invests USD 100,000 in the information security bond shown in Table 7, then the investor will earn returns based on the average performance of firewalls against the 'UVW' type attacks as shown in Table 8.

TABLE VIII. RETURNS ON INVESTMENT OF USD 100,000 FROM INFORMATION SECURITY BONDS.

Result	Performance Score on 31/Dec/2017	Return	Returns to Investors
Performance unchanged	10	10%	$10,000
Performance improves by 10%	11	20%	$20,000
Performance improves by 20%	12	25%	$25,000
Performance improves by 30%	13	30%	$30,000

As shown in Table 7 and Table 8, information security stakeholders can coordinate their efforts in strengthening the information security ecosystem and can reap significant profits from the financial instruments.

VII. An Evaluation of ISFI

The artifact evaluation consists of three sub-activities [16]. The first activity 'analyze context', analyzes and describes the context of evaluation. The second, 'select goals and strategy', is not only about deciding the goals and strategy for the evaluation but also about selection of research strategy and methods. The third sub-activity, designs the evaluation study and then executes the same.

Figure 6 shows the artifact (ISFI) evaluation process.

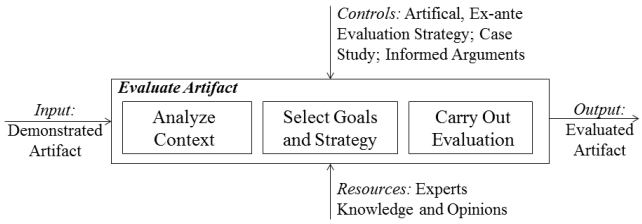


Figure 6. ISFI Evaluation Process (adapted from [16]).

- *Input:* describes the knowledge or object which is the input to evaluation activity.
- *Output:* describes the knowledge or object which is the outcome of the evaluation activity.
- *Controls:* describes the knowledge that is used for evaluation activity, including evaluation strategies.
- *Resources:* describes the knowledge which is used as the basis for the evaluation exercise, i.e., the knowledge base.

A. Analyze Context

The first sub-activity is 'Analyze Context', and it primarily identifies the constraints in the evaluation environment [16]. The main constraint in the evaluation of ISFI are the technological, financial, legal and time constraints.

B. Select Goals and Strategy

The second sub-activity 'Select Goals and Strategy' is based on the evaluation context. The goals selected are to evaluate the ISFI against the identified requirements, and its usefulness in addressing the previously identified problems, using formative evaluation. One of the six evaluation types stated in [24] is to 'Comparison'. It implies that the artifact is not evaluated in isolation indeed it is studied in comparison to other artifacts meant for the same or similar purpose. Therefore, where necessary, we have compared our artifact with cyber-insurance products. The evaluation strategy selected is artificial, and ex-ante. The ISFI is evaluated using an 'informed argument' strategy. The formative evaluation is chosen because the results of the evaluation may lead to several iterations before the design of ISFI is finalized.

C. Carry Out Evaluation

The evaluation consists of two parts. First, to evaluate the artifact against the ISFI requirements. Second, to evaluate the usefulness of ISFI.

1) Evaluation against ISFI Requirements: The following evaluation is only for the functional requirements of ISFI. As the usability requirements (and their achievements) are dependent upon the mechanism facilitating the trading/investment in the instrument, thus the usability requirements should be evaluated with respect to the platform. Therefore, the evaluation of usability requirements is beyond the scope of this evaluation.

 ISBN: 978-1-61208-427-5

TABLE VII. AN EXAMPLE APPLICATION OF INFORMATION SECURITY FINANCIAL INSTRUMENTS.

Issuer	Information Security Product Vendors Association
Objective of the Funding	To improve the performance of firewalls developed by Europe based companies to defend against UVW type attacks in two years
Benchmark Measurement Criteria	Firewall performance index against UVW type attacks
Total Funding Required	Amount : 1,000,000; Currency : USD
Project Start Date	01-Jan-2016
Project End Date	31-Dec-2017
Information Security Financial Instrument Type	Information Security Bond
Transferable Instrument	Yes, only to verified traders/investors registered with the clearing house/information security prediction market
Decision Criteria	The average performance of firewalls developed by European companies against the UVW type attacks must improve by at least 10% before the project end date.
Initial Benchmark Value	Let us assume that there are three firewalls developed by European companies and providing defense against UVW type attacks and each has a global market coverage of at least 30% *Firewall 1 :* Average Performance Index between 01-Jan-2015 to 31-Dec-2015 is 10 *Firewall 2 :* Average Performance Index between 01-Jan-2015 to 31-Dec-2015 is 8 *Firewall 3 :* Average Performance Index between 01-Jan-2015 to 31-Dec-2015 is 12 Average Performance Index for the three firewalls between 01-Jan-2015 to 31-Dec-2015 is 10
Minimum Investment Required	Amount : 10,000; Currency : USD
Eligible Investors	Information Security Product Vendors, Cyber-Insurance Providers, Reinsurance Providers, Information Security Researchers, Security Industry Consortium, Investment Managers, Product Users
Independent Third Party Quality Rating Required	Yes, for credit rating of issuer
Independent Third Party Verification Required	Yes, for the performance evaluation of the firewall
Management Fee	2%
Know Your Trader/Investor Required	Yes, verification of personal and minimal financial background of participants.
Return Structure	Mixed (Tiered and Incremental) Base Return: 10% irrespective of firewall performance index after two years. The incremental returns are linked to the actual performance outcome of firewalls developed by European companies as below: *(i) Base Yield = 10%*; *(ii) 10% above the reference = 20%* *(iii) 20% above the reference = 25%*; *(iv) 30% above the reference = 30%* For performance between the above tiers, returns are calculated on pro-rata basis.
Pay-Off Horizon	07 days from the project end date
Bonus Payment	Yes
Trigger for Bonus Payment	As specified in return structure section

- *Coordinated Efforts and Investments:* In the absence of efficient and effective cyber-insurance markets, incentives to engage in prevention and insurance are reduced. In the absence of an effective cyber-reinsurance market, the government is expected to become the financier of huge systemic losses [25]. Alternatively, governments can encourage the information security stakeholders to engage in risk financing through ISFI that, in turn may cover the risk exposures. ISFIs can provide a project based approach to manage systemic risk through mitigation or risk transfer, will reduce specific 'threat/vulnerability' exposure and may lead to better risk management practice, thus strengthening the information security ecosystem.
  ISFIs can be used as a mechanism to combine risk exposures spread over several information security defense products. This can be achieved by pooling the systemic risk exposure across the product types to provide a natural first line of defense by engaging the stakeholders in coordinating the efforts to strengthen the information security ecosystem. It may also provide a scale economics to finance risk arrangements in international information security markets.
  As demonstrated in Section 6, an ISFI (bond) is used to engage various information security stakeholders particularly the companies developing firewalls in Europe and providing a defense against 'UVW' type attacks. Through the information security bonds, these firewall developers can invest in coordinated efforts to improve the performance of their firewalls to defend against the 'UVW' type attacks. After the end of the project, the project (performance) data can be used to market their firewalls as a (more) effective product, thereby better positioned against the firewall developers from other regions.
- *Tied Returns:* The combination of higher event frequency and extended exposure increase the potential damages. Despite the growing information security risk exposure, cyber-insurance markets are not mature and effective enough to counter the risks. Therefore, most of the organizations (and individual users) are exposed to information security risks and do not have (adequate) financial coverage. Given the fact, a 'proactive' use of alternative risk management mechanism may be worth considering.
  ISFI can provide a 'proactive' mechanism for information security risk management. To achieve this, returns on ISFI are tied to the achievements of pre-specified performance or results expectations. These payment triggers are clearly defined, objectively measurable and independently verifiable.
  As demonstrated in Section 6, ISFI (bonds) are issued with an objective of improving the performance of firewalls developed by Europe-based companies, and the returns are tied to the 'firewall performance index'.
- *Accountability:* Researchers, industry practitioners and the legal fraternity have been arguing for a very long time over the issue of software 'bug/vulnerability' liability [26]–[29]. However, the discussion on the topic remains inconclusive. ISFI aims to target this issue by fixing the accountability on the stakeholders (such as on a product vendor) to fix the bug or to reduce the number of bugs in a piece of software in lieu of returns tied to the achievement of the same.

 ISBN: 978-1-61208-427-5

As demonstrated in Section 6, the 'information security product vendors association' is the issuer and owns the accountability to achieve the desired performance of firewalls against the specified attacks. A failure to achieve the desired performance is likely to result into losing the competition to others, facing the opportunity cost, and so on.

- *Return Structure:* ISFI provides a variety of return structure depending on the objective of the issuer and other security stakeholders. The payment triggers are linked to the achievement of pre-specified performance or observable results.
  As demonstrated in Section 6, the information security bond provides a mixed return structure. It incentivizes achievement of as high as the possible performance of the firewall, so as to earn maximum possible returns based on the incremental tier structure.
- *Designed as Equity, Debt or Convertible Instrument:* ISFI are tailor made products to address the specific (underlying) problems or objectives. Therefore, to cater to the variety of objectives, events, and needs of security stakeholders, ISFI have the flexibility to be designed as equity, debt, and convertibles.
  As demonstrated in Section 6, ISFI is a bond type instruments designed to source funds to improve the performance of firewalls. Similarly, equity and convertible types of instruments can be drawn to meet specific functional requirements.

2) Evaluation against ISFI Usefulness: The evaluation of ISFI against its usefulness in dealing with the problem of information asymmetry, negative externality, and free riding is as follows:

- *The Market for Lemons:* ISFI targets the problem of information asymmetry where sellers have no (or minimal) incentive in producing robust products. ISFI can be used as a method to prove the performance of the target products.
  For instance, as demonstrated in Section 6, the firewall vendors can use the performance data of the information security bond to prove that their firewalls are better than the other (lemons) firewalls providing defense against the UVW type attacks. This works as a product rating or quality assurance for the buyer. In such a scenario, firewall vendor with proven performance of the firewall may charge a higher price than its competitors (i.e., lemons).
- *The Market for Insurance:* ISFI can be used to address the problem of 'market for insurance'. In such a scenario, customers willing to purchase cyber-insurance policies can prove the resilience of their information security defense system by using the software, hardware and practices & policies, which have achieved a certain level of performance as exhibited through ISFI. This will create a level of confidence in the cyber-insurance provider, and the insurance buyer can negotiate for a lower premium or inclusion of certain other risk coverage.
  For instance, as demonstrated in Section 6, a user using one of the firewall which achieved the desired performance as per the information security bond can claim a better protection against UVW attacks compared to those who are using other firewalls and thus negotiate for a lower premium.
- *The Tragedy of Commons:* ISFI targets the problem of negative externality and free riding by incentivizing the coordinated efforts of various stakeholders. ISFI encourages investments in robust security products, and visibility of quality and performance of these products will lead to natural robustness in the ecosystem.
  For instance, as demonstrated in Section 6, if the information security bonds lead to achievement of desired performance of the firewalls then the government can bring in mechanisms like tax credit [30] to encourage usage of proven security products and wide acceptance of these products will help in eliminating the issue of negative externality and free riding from the information security ecosystem.

VIII. Conclusion and Future Work

In this paper, we have shown a need for an alternative risk management method. We have identified a set of requirements for Information Security Financial Instruments (ISFI) which can be used as an alternative risk management mechanism to incentivize coordinated efforts by security stakeholders in strengthening the information security ecosystem. We have designed the ISFI and demonstrated its application with an imaginary case of improving firewall performance. Then, we analyzed the ISFI against the set of functional requirements and its usefulness in addressing various economic problems prevalent in information security domain. In our analysis, we found that the ISFI meets all the functional requirements for the instrument. Also, on the issue of addressing the problems of information asymmetry, negative externality and free riding, ISFI can be highly useful. However, as our analysis is based on 'informed argument' evaluation method, the evaluation faces a high risk of false positives.

There are three limitations in the paper: (i) ISFI is demonstrated with a 'bond' type instrument only. Application of equity and convertible type instruments are not presented and left to the future work. (ii) Our evaluation of ISFI is only for the functional requirements; however usability requirements may have a significant impact on success or failure of the instrument, and this is left for future work. (iii) We have demonstrated and evaluated the ISFI based on an imaginary case; however there could be several constraints when implementing the instrument in a naturalistic setting.

References

[1] L. A. Gordon, M. P. Loeb, and L. Zhou, "The impact of information security breaches: Has there been a downward shift in costs?" Journal of Computer Security, vol. 19, no. 1, Jan. 2011, pp. 33–56.

[2] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, "The economic impact of cyber-attacks," Government and Finance Division, Congressional Research Service,, CRS Report for Congress Order Code RL3233, April 2004.

[3] A. Smith. Share prices are rarely hit hard by cyber attacks. Financial Times. http://www.ft.com/intl/cms/s/0/348d7f1a-417e-11e3-9073-00144feabdc0.html. [retrieved: Apr, 2015]

[4] D. Y. Emily Glazer. J.p. morgan says about 76 million households affected by cyber breach. Wall Street Journal. [Online]. Available: http://www.wsj.com/articles/j-p-morgan-says-about-76-million-households-affected-by-cyber-breach-1412283372 [retrieved: Apr, 2015]

 ISBN: 978-1-61208-427-5

[5] J. Tom Huddleston. What you need to know after sony's hacker attack. Fortune. [Online]. Available: http://fortune.com/2014/12/03/need-to-know-cyber-attacks/ [retrieved: Apr, 2015]

[6] H. Kuchler and A. Raval. Target data theft sounds wake-up call for retailers. Financial Times. http://www.ft.com/intl/cms/s/0/7d5f28bc-7d81-11e3-81dd-00144feabdc0.html. [retrieved: Apr, 2015]

[7] G. A. Akerlof, "The market for 'lemons': Quality uncertainty and the market mechanism," Quarterly Journal of Economics (The MIT Press), vol. 84(3), 1970, pp. 488–500.

[8] R. Anderson, T. Moore, S. Nagaraja, and A. Ozment, Algorithmic Game Theory, 2007, ch. Incentives and Information Security, pp. 633–649.

[9] G. Hardin, "The tragedy of the commons," Science, vol. 162, 1968, pp. 1243–1248.

[10] E. R. McNicholas. Cybersecurity insurance to mitigate cyber-risks and sec disclosure obligations. The Bureau of National Affairs, Inc. http://www.bna.com/cybersecurity-insurance-to-mitigate-cyber-risks-and-sec-disclosure-obligations/. [retrieved: Apr, 2015]

[11] Zurich American Insurance Company vs Sony Corporation of America, no. No. 651982/2011. NewYork Supreme Court, Jul 2011.

[12] R. King. Cyber insurance capacity is very small: Aig ceo. CIO Journal. Wall Street Journal. [Online]. Available: http://blogs.wsj.com/cio/2015/04/02/cyber-insurance-capacity-is-very-small-aig-ceo/ [retrieved: Apr, 2015]

[13] M. Lelarge and J. Bolot, "Economic incentives to increase security in the internet: The case for insurance," in INFOCOM 2009, IEEE. IEEE, 2009, pp. 1494–1502.

[14] Z. Yang and J. C. Lui, "Security adoption and influence of cyber-insurance markets in heterogeneous networks," Performance Evaluation, vol. 74, no. 0, 2014, pp. 1 – 17.

[15] K. J. Arrow, "Uncertainty and the welfare economics of medical care," The American Economic Review, vol. 53, no. 5, 1963, pp. 941–973.

[16] P. Johannesson and E. Perjons, An Introduction to Design Science, 1st ed. Springer International Publishing, 2014, iSBN: 978-3-319-10631-1.

[17] M. Rothschild and J. Stiglitz, "Equilibrium in competitive insurance markets: an essay on the economics of imperfect information," Quarterly Journal of Economics, vol. 90, 1976, p. 629.

[18] WEF and Partners, "Partnering for cyber resilience: Risk and responsibility in a hyperconnected world - principles and guidelines," World Economic Forum, Tech. Rep. Ref. 270912, March 2012.

[19] WEF and Partner, "Risk and responsibility in a hyperconnected world," World Economic Forum in collaboration with McKinsey & Company, Tech. Rep., Jan 2014.

[20] "Purefunds ise cyber security etf," Pure Funds, Tech. Rep., 2014. [Online]. Available: http://pureetfs.com/etfs/hack.html

[21] "Uk cyber vulnerability index 2013," KPMG Consulting, Business and industry issue, May 2014.

[22] "Global cybersecurity index," International Telecommunication Union and ABI Research, Tech. Rep., 2014.

[23] Index of cyber security. [Online]. Available: http://www.cybersecurityindex.org/ [retrieved: Apr, 2015]

[24] J. Venable, J. Pries-Heje, and R. Baskerville, "A comprehensive framework for evaluation in design science research," in Design Science Research in Information Systems. Advances in Theory and Practice. Springer, 2012, pp. 423–438.

[25] A. Gray. Government resists calls to fund backstop for cyber disaster losses. Financial Times. [Online]. Available: http://www.ft.com/cms/s/0/7f9d8326-d096-11e4-a840-00144feab7de.html [retrieved: Apr, 2015]

[26] R. Clarke, "Who is liable for software errors? proposed new product liability law in australia," Computer Law & Security Review, vol. 5, no. 1, 1989, pp. 28 – 32.

[27] J. Armour and W. S. Humphrey, "Software product liability," School of Law, University of Pittsburgh and SEI, Carnegie Mellon University, USA, Tech. Rep. CMU/SEI-93-TR-13, ESC-TR-93-190, Aug 1993.

[28] J. Goodchild. Security experts: Developers responsible for programming problems. [Online]. Available: http://www.csoonline.com/article/2124824/malware-cybercrime/security-experts--developers-responsible-for-programming-problems.html [retrieved: Apr, 2015]

[29] M. Masnick. U.k. court says software company can be liable for buggy software. [Online]. Available: https://www.techdirt.com/blog/innovation/articles/20100513/0053499408.shtml [retrieved: Apr, 2015]

[30] 113th Congress (2013-2014), Ed., Cyber Information Sharing Tax Credit Act, no. S.2717, Senate of the United States. Senate - Finance, July 2014. [Online]. Available: https://www.congress.gov/bill/113th-congress/senate-bill/2717/text

 ISBN: 978-1-61208-427-5

You Are Who You Know

Leveraging webs-of-trust for authentication in identity federations

Bob Hulsebosch, Arnout van Velzen,
Maarten Wegdam & Martijn Oostdijk
InnoValor
Enschede, The Netherlands
e-mail: bob.hulsebosch@innovalor.nl

Remco Poortinga-van Wijnen, Joost van Dijk
SURFnet
Utrecht, The Netherlands
e-mail: remco.poortinga@surfnet.nl

***Abstract*—Digital identity assurance emerges from two aspects: the strength of the authentication solution, or how you identify yourself towards an online service, and quality of the identity proofing and registration process, or how the authentication solution was issued to you. A reliable registration process, however, is often expensive. For example, it may require the establishment of a registration desk, which is not very user friendly as it demands much effort on the part of the user. This paper investigates the feasibility of using webs-of-trust for reliable identity proofing in digital authentication. Webs-of-trust entail communities of people that trust each other, i.e. utilizing social contacts to confirm people's identities. A functional decomposition of an attestation service and protocol for web-of-trust enhanced authentication are provided. A prototype for an attestation service was developed as a proof-of-concept, leveraging LinkedIn as a web-of-trust, and evaluated by users. Finally, characteristics of using web-of-trust for authentication assurance are discussed and a Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis was conducted. Key findings are that while webs-of-trust provide an interesting alternative mechanism for identity proofing that may have merit in use cases where no more efficient registration processes are available, its implementation is complex and mainly challenged by usability.**

Keywords-authentication; web-of-trust; level of assurance; attestation service; identity proofing.

I. INTRODUCTION

Authentication refers to an online process where an entity's identity is verified, typically by providing evidence that it holds a specific digital credential. The strength, or degree or reliability, of the authentication solution is usually expressed in terms of Levels of Assurance (LoA). Two factors are essential in the determination of the LoA [1][2]:

1. The quality of the registration process, i.e., of the identity proofing, registration, and the delivery of credentials that are bound to the registered identity.
2. The strength of the authentication process to establish that a user is who he/she claims to be, which in turn mainly depends upon the strength of the authentication credential.

There is an increasing need for two-factor authentication solutions with cost efficient identity registration. The use of second factor authentication credentials is growing but lack reliable registration processes by which to link a physical person to his/her digital identity information and to his/her authentication credentials during enrolment weaken the overall authentication strength. If this is done poorly, there is little or no assurance that the person using that credential is who he/she claims to be.

Different registration processes and mechanisms apply to identity vetting, proofing, credentialing and linking, and result in different assurance levels. An applicant may appear in person to register or may register remotely. In person registration provides reliable identity proofing, but is expensive (typically from €10 upwards) and not very user friendly (e.g., going to a registration office). Remote registration generally relies on the availability of trusted sources to cross-reference and validate the provided assertions such as name, home address, age, e-mail address, and photo. Remote registration is relatively cheap, but is vulnerable to threats and technically complex. This often leads to weak binding between the user, his authentication credential, and his digital identity. Consequently, the authentication LoA will be low.

An innovative approach to achieve a higher registration LoA, without the cost and overhead of physical registration, is based on the concept of web-of-trust. Using webs-of-trust the authenticity of the binding between an authentication solution and its owner is established via third party user attestations. For instance, if person A claims that user B is using a particular digital identity, it could provide extra confidence for the service provider to allow access to resources that require a certain level of authentication assurance. When Person C also confirms that this digital identity is used by person B, this further increases trust in the digital identity of B. This mechanism can be considered "crowdsourcing of trust". The relations between person A, B, and C, i.e. they share the same social or professional context, could be used to further enhance the level of the authentication assurance.

Particularly in the context of research groups or virtual organizations in which users commonly know each other, such web-of-trust-based authentication LoA enhancement could be executed in an efficient manner. Moreover this approach also promises to capitalize on authentication

 ISBN: 978-1-61208-427-5

functionality provided by social networks such as LinkedIn, Facebook and Google in higher education and research environments. The registration LoA part of the authentication solutions provided by these networks is relatively weak (LoA 1) despite the fact that an increasing number of them are using two-factor authentication (LoA 2 or higher). Web-of-trust based LoA enhancement could help increasing the registration LoA part of these providers and thus could help in increasing the overall LoA.

The objective of this study is to determine the feasibility of using webs-of-trust to enhance the level of the authentication assurance, i.e., having your social connections vouch for your identity. In a way, this implies crowdsourcing assurance for identity verification.

The structure of the paper is as follows. Section II provides some background on webs-of-trust. Section III describes the functional decomposition for an attestation service that enables web-of-trust-based authentication. A protocol for leveraging web-of-trust for authentication and its implementation are described in Section IV. A user evaluation of the prototype that was developed based on this protocol as a proof-of-concept is described in Section V. A Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis is given in Section VI. Related work is briefly touched upon in Section VII. Finally, Section VIII draws conclusions and provides an outlook for future research.

II. WEB-OF-TRUST

The web-of-trust concept is based on the idea of decentralized trust and social networks. It is used in Pretty Good Privacy (PGP) [3] as an alternative to the centralized trust model that is the basis of a public key infrastructure. In a web-of-trust, each user of the system can choose for himself whom he elects to trust, and who not. Instead of trusting a single entity to validate identities, you validate the identities of the people you know and export this information to a public database. Then, you rely on friends to vouch for the people they know, and those friends to vouch for still more people, and so on until you create a trust chain between any two arbitrary identities. This approach avoids the inherent problems of central authorities, but in practice it is rarely used due to usability issues of tools involved and a lack of user incentive.

A successful web-of-trust should likely be built much like an online social network to obtain the shared experience information for certification, which is a model that hundreds of millions of people all over the world are already comfortable with using. As such, the web-of-trust model can be used to establish the authenticity of the binding between an authentication solution and its owner via third party user attests. Instead of building a whole new web-of-trust, existing trust infrastructures such as PGP, FoaF [4], identity federations, social or professional networks should be readily reused to enhance the registration component of the overall LoA.

LinkedIn is the world's largest business social networking site. One purpose of the site is to allow registered users to maintain a list of contact details of people with whom they have some level of relationship, called Connections. Users may invite anyone (whether a site user or not) to become a connection. LinkedIn provides an interface to obtain basic profile information of users. Information about the connected users in the LinkedIn network of a user can be collected as well. The availability of the information depends on the privacy policy of the connected user. As such, LinkedIn provides sufficient information to determine a reliable set of users that may enhance the level of assurance in someone's identity. The same holds for similar social networks such as Google+, Orkut, and Facebook.

Potentially, the web-of-trust approach combines the best of remote and physical registration practices. There is no need for a physical registration desk as other users in the web-of-trust take over the responsibility to identify users. Confidants in the web-of-trust may use physical presence, phone or email practices for this purpose. However, the attestations from the web-of-trust somehow need to be related to the claimant's digital identity. This needs to be catered for by some kind of attestation service.

III. FUNCTIONAL DECOMPOSITION

Three user-roles can be distinguished in a web-of-trust based authentication scenario:

1. An Asker that wants to use the Attestation Service to enhance assurance of his identity.
2. A Helper that attests for the Asker's identity.
3. A Moderator that wants to have someone's identity (i.e., an Asker) attested.

A functional decomposition results in a number of building blocks that are required to realize web-of-trust based authentication. These are shown in Figure 1.

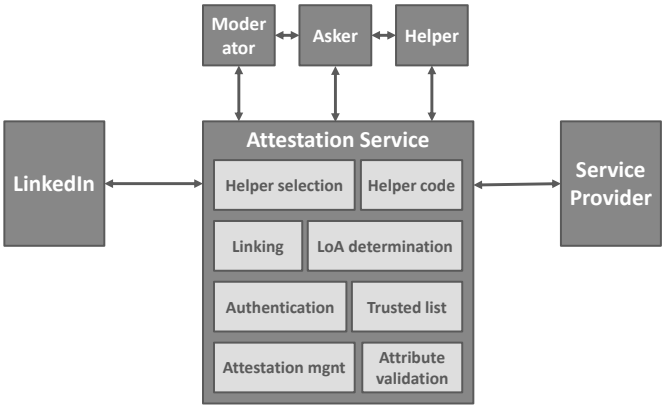


Figure 1. Functional decomposition.

The need for an Attestation Service that facilitates and coordinates the web-of-trust based enhancement of the authentication solution is obvious. Specific functionalities of such an Attestation Service are:

- Authentication of the users (Asker, Helper, Moderator). Authentication could be done in federated manner, via social logon, or locally. Ideally, the Asker as a strong authentication credential with a low LoA due to unreliable registration of the credential to the user's identity.

 ISBN: 978-1-61208-427-5

- Helper selection: who are the best Helpers to attest for the Asker's identity? Candidate Helper selection should be such that it mitigates risks related to herd behavior and fake accounts. Ideally, Helpers come from multiple webs-of-trust and have varying relationships with the Asker (e.g., friend/colleague, recent/longtime, etc.).
- Helper code. The Attestation Service needs to be sure that the selected Helpers are indeed the ones that login to vouch for the Asker's identity. One way to achieve this is by generating a random code that is passed to Helpers that they then have to enter to verify their attests are bona fide.
- Linking of social networks to an Asker or Moderator, i.e., giving the Attestation Service access to the LinkedIn social graph data. This enables the Attestation Service to select meaningful Helpers from the social network. Commonly, users should be able to link their social network accounts to the Attestation Service.
- LoA determination based on Helper attestations. Aspects that could be taken into account are: the number of Helpers, the LoA of Helpers, and the number of invited Helpers that did not vouch. The outcome of the LoA is communicated to the Asker and the service provider.
- Trusted list: establishing a list of trusted Helpers from which Helpers will be primarily selected against the social graph of the Asker. In case of the moderator-scenario, the list consists of the Helpers from the Moderator's social network.
- Attestation management, i.e., keeping track of the attestations given by Helpers, giving feedback to the Moderator or the Asker, asking Helpers to become trusted Helpers.
- Attribute validation could be optional functionality of the Attestation Service. The Attestation Service may ask the web-of-trust to verify self-asserted personal attributes of the user such as a telephone number, age, or address.

IV. PROTOCOL AND IMPLEMENTATON

A. Protocol

The following protocol for web-of-trust enhanced authentication has been implemented in the proof-of-concept:

Step 0: *Building Trust List, Moderation:* A list of trusted potential Helpers may need to be created. A Moderator may make an attestation request for a particular Asker.

Step 1*: Registration of Asker.* Asker registers at the Attestation Service by logging in with his/her federated identity and requests enhancement of authentication. The response of the identity provider contains identity information of Asker. The information at least contains a LoA attribute and value and Asker's federated user identity identifier. Asker is asked to link his/her federated institution account to, e.g., his/her LinkedIn account by logging in with his/her LinkedIn credentials.

Step 2*: Web-of-trust scoping.* The Attestation Service determines who is able to vet for Asker's identity by imposing its trust requirements on the available web-of-trust of Asker. Once the web-of-trust has been determined (in this case LinkedIn) the Attestation Service can start selecting suitable Helpers. Subsequently, Asker is given a vouching code and is asked to contact the Helpers by phone or physically and pass them the code. The use of e-mail is prohibited or deprecated; Asker has to affirm that he/she will adhere to this policy. Asking too many Helpers will burden the Asker as he/she has to contact them.

Step 3*: Passing of vouching code.* Asker calls or meets Helpers and tells them the vouching code. During the phone call or meeting, the Helpers implicitly authenticate the Asker (e.g., via voice or face recognition).

Step 4*: Helper vouching.* The Helper logs in to the Attestation Service with his/her federated identity credentials. The authentication solutions he/she is using must have an equal or higher assurance level than Asker's current level. After successful authentication, the Helper states which Asker he/she wants to vouch for, and the Attestation Service asks the Helper to enter the vouching code. The Attestation Service then validates if the Helper is indeed one of the selected Helpers. If this is the case it asks the Helper to confirm that he/she vouches for Asker's identity. Optionally the Attestation Service may show Asker's personal attributes and asks Helper to validate them. Afterward the Helper logs out. Helper validation can be done in several ways. For instance, the Attestation Service might compare the attributes provided by the identity provider during authentication with those of the selected Helpers from Asker's social network. They should overlap. Another approach is to send the Helper an email with a specific code. The Helper must enter the code together with the vouching code.

Step 5*: LoA determination.* The Attestation Service updates the LoA of Asker based on the number of Helper attestations and their LoA. Mapping web-of-trust-based LoAs to existing frameworks for LoAs like ISO29115 [1] or STORK [2] is not possible; these frameworks do not take web-of-trust mechanisms into account. Consequently, we defined our own web-of-trust-based LoA-framework consists of three levels:

1. WoT LoA1: equal to LoA1 of STORK or ISO29115.
2. WoT LoA2: requires a
   a. minimum of 5 Helpers with LoA1 / WoT LoA1, or
   b. minimum of 3 helpers with LoA2 / WoT LoA2
3. WoT LoA3: requires a
   a. minimum of 8 helpers with LoA2 / WoT LoA2, or
   b. minimum of 5 helpers with LoA3 / LoA4 / WoT LoA3

Also, the number of invited Helpers that did not vouch should be taken into account. These may be considered as 'negative vets'. They have a negative effect on the new LoA. A simple algorithm is to multiply the new WoT LoA with the percentage of positive vets. Note that this is an initial definition of the LoAs, just to get an impression of what it means to step-up to a higher level. The Asker is notified by the Attestation Service about the new LoA, i.e., attestation status.

Step 6*: LoA communication.* Next, Asker can go to a service provider and authenticate himself/herself using his/her federated identity. Multiple solutions are possible for

 ISBN: 978-1-61208-427-5

the communication of the LoA. One possible solution is that the identity provider authenticates Asker at e.g. LoA 1 and communicates this to the service provider. The service provider decides that this is not sufficient and makes a LoA attribute validation request at the Attestation Service. The Attestation Service returns a LoA 2 attribute. This convinces the service provider to allow Asker access to the service. Another solution is that the Attestation Service becomes the (new) identity provider for the Asker, authenticates him/her and communicates the LoA to the service provider. This implies that the Asker must be able to select the Attestation Service as her preferred identity provider.

The different steps are illustrated in Figure 2. The protocol is inspired by the work of Brainard on using vouching by which helpers leverage their strong authentication in order to assist another user, the asker, to perform emergency authentication in case of loss of a second authentication token [5].

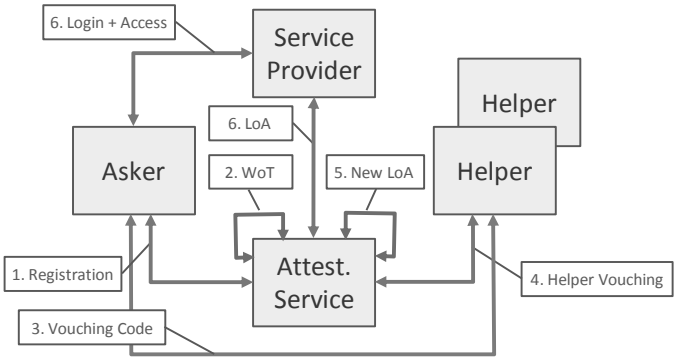


Figure 2. Web-of-trust protocol flow.

B. *Implementation scenario*

A proof-of-concept Attestation Service has been developed. It models a web service for step-up authentication for access to a shared research environment. The Attestation Service allows users to login with a local username and password combination. This can easily by extended to other federated authentication or social login solutions. In case of federated authentication, the attributes that are provided by the identity provider during authentication at the Attestation Service could be used for validation purposes. Furthermore, the Attestation Service offers the user the opportunity to get attested and link the identity provider account to her LinkedIn account.

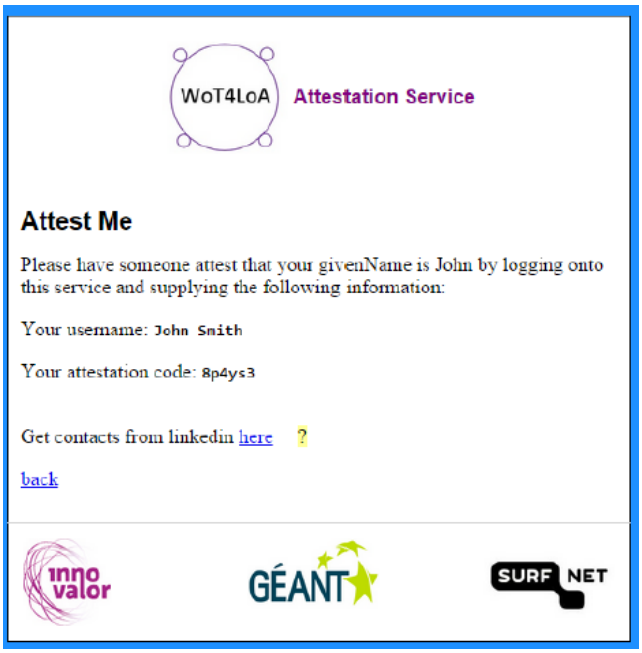


Figure 3. Asker wanting to be attested by Helpers.

The latter provides the Attestation Service the ability to randomly select 5 Helpers from the LinkedIn web-of-trust of the Asker or to select the Helpers from its own list of trusted Helpers. Helpers can put themselves on this trust list by sharing their LinkedIn contacts. Upon submitting an attestation request, the Asker is presented a vouching code that is alphanumeric and consists of five characters, with the instruction to approach the five Helpers (but not by e-mail). Helpers should login to the Attestation Service with their federated account and fill in the vouching code in order to verify the Asker's identity. When all five Helpers have vouched, the hypothetical LoA of the Asker is stepped-up from 1 to 2, granting the Asker access to the concept shared research environment. A Moderator may also request an attestation for an Asker, view the progress of attestations or set his own LinkedIn contacts as the trust list to select Helpers from.

To give an impression of the proof-of-concept several screenshots are shown in Figure 3 and Figure 4.

 ISBN: 978-1-61208-427-5

Figure 4. Helper attesting Asker.

V. EVALUATION

To test the concept of web-of-trust the proof of concept was evaluated by two separate user groups via role-playing scenarios. In addition to specific questions, remarks of the participants as well as non-verbal communication were noted by the observers for evaluation of the prototype.

The outcomes of these two user evaluation tests show that usability is a critical factor for the success of web-of-trust enhanced authentication. Also, the concept is relatively difficult to explain to users. Furthermore, not everyone actively uses social media, e.g., exemplified by not knowing username and password. This could lead to frustration on the part of all three roles (Asker, Helper, and Moderator). The users experienced barriers to contact Helpers and motivating them to provide an attestation. Moreover, it is likely that situations will occur wherein Askers are unable to reach Helpers, e.g., because they do not possess sufficient contact details. Similarly, non-response handling of Helpers could be problematic, since a Helper response cannot be guaranteed. The reliance on others may obstruct or delay authentication and access. So, in order to achieve successful and timely attestation, the whole attestation process should be strictly guided by the Attestation Service. For instance, communication between Helper, Moderator, Asker and Attestation Service could be automated or manually performed through a host of channels, albeit each with their own considerations and trade-offs in terms of responsiveness and 'social pressure'.

According to the evaluation results, there are also trade-offs inherent to Helper selection; not always suitable Helpers were selected. Helper selection is dependent on the information provided by the web-of-trust and the quality of the reasoning algorithm for selecting them. A better selection may be possible if more information is available from the social network used as a source, e.g., the number of likes and comments (cross-) posted on Facebook or the duration of a LinkedIn connection. This information is typically not available to applications outside of the social network itself. Moreover, users may be uncomfortable making the information available to the Attestation Service, as was witnessed by the comments during the prototype evaluation.

Which social network is most appropriate to get attestations from depends on the type of service to be accessed by the Asker. A work-related service would favor the use of a professional social network such as LinkedIn to get attestations from; a leisure or e-commerce type of service might benefit from attestations from the Facebook web-of-trust.

The challenges of automatically selecting the 'right' social network and (then) the 'best' Helpers can be circumvented by restricting the context and work flows for this approach to only Moderator-initiated attestation. The selection of network and Helpers can then conceivably be done by the Moderator, although that does raise the question what additional benefit this approach has if the moderator already has enough information and knowledge to do that selection in the first place (i.e., is attestation really still needed in that situation?). Conceivably removing the social network from the equation altogether and allowing the Moderator to appoint 'delegated Registration Authorities' may work better in those situations.

VI. SWOT ANALYSIS

This section discusses the strengths, weaknesses, opportunities and threats (SWOT) of web-of-trust based authentication approaches, followed by a feasibility analysis to determine whether threats can be mitigated and opportunities leveraged by using the strengths and eliminating the weaknesses.

A. *Strengths*

1) *Cost efficient*

The web-of-trust approach combines the best of remote and physical registration practices. There is no need for an expensive physical registration desk as other users in the web-of-trust take over the identification task, which reduces costs of enrolling strong authentication.

2) *Less intrusive for the user*

Potentially it reduces the intrusiveness for the user as it replaces the cumbersome physical registration overhead by more natural Asker-Helper interactions. Askers, however, may be reluctant to ask a Helper they haven't seen or spoken for quite some time to attest for their identity.

3) *Easy integration in existing federation infrastructures*

The Attestation Service can be easily integrated in an existing identity federation infrastructure. It can leverage the existing federated trust fabric for selecting reliable helpers. The Attestation Service can be positioned as an attribute provider for federated service providers. It can make

assertions about the LoA level of the user. Moreover, contrary to other approaches - such as PGP or FOAF - there is no need for specific client software at the user side.

B. Weaknesses

1) Reliability

ENISA has summarized the possible threats to reputation-based systems. Examples of threats are whitewashing attacks, Sybil attacks, impersonation and reputation theft, bootstrap issues related to newcomers, extortion, denial-of-reputation, ballot stuffing and bad mouthing, collusion, repudiation of data and transaction, recommender dishonesty, privacy threats for voters and reputation owners, social threats such as discrimination or risk of herd behavior, attacking of the underlying infrastructure and the exploitation of features of metrics used by the system to calculate the identity assurance [6]. Most of these threats are also applicable to web-of-trust based authentication. Though the proposed approach does not mitigate all of these threats, their impact is largely influenced by the quality of the Attestation Service's reasoning algorithm. Moreover, using social networks as a web-of-trust for identity attestations makes it more difficult to spoof the system by creating false identities or colluding in groups.

It is relatively easy for an Asker to create multiple LinkedIn, Facebook or Google+ accounts under fake identities and establish via these accounts a web-of-trust of LinkedIn connections or Facebook or Google+ friends (i.e. Sybil attack). This threat is largely mitigated by the fact that the Attestation Service determines the Helpers. Additionally, it can be required for Helpers to have a higher LoA than the Asker; this makes it more difficult to create false Helpers.

False identities can be detected by a relatively poor social ranking. Either they remain disconnected or are connected to a relatively isolated group of 'old friends'. Large-scale analysis of social networks can uncover at least some forms of group collusion. For example, web pages colluding to alter their search engine ranking by linking to one another can be identified and removed if they all have a similar number of links [7]. Alternately, collusion could alter the relative abundance of motifs (small sub-graphs), arousing suspicion if it differs significantly from that of social networks in general [8].

Similarly, herd behavior due to social pressure can be circumvented in a similar manner by selecting Helpers from different webs of trust. Reliable selection functionality may prevent the situation of a group of attackers that collaborate to boost their identity assurance via false attestations.

Services exist that analyze many sources including social networks such as Facebook, Google+, LinkedIn and Twitter to verify cyber identities in real-time. These services are able to detect fake accounts and corresponding identities. An example is Trulioo that offers a service that analyses Facebook profiles and determines whether they're likely to be spoof accounts [9].

However, not all risks can be mitigated completely. Given this weakness, the web-of-trust approach may not be suitable to achieve the highest LoA (i.e., 4), but certainly has the potential to achieve LoA 3.

2) Liability

Another weakness is related to liability. The Attestation Service becomes the authority regarding the authentication LoA of the user. Its owner can, however, not easily be made liable for its LoA claims. The relying service provider has to trust the web-of-trust based LoA claims of the Attestation Service. The fact that both the Attestation Service provider and the relying service provider are in the same federation may help establishing this trust. Additionally a mechanism could be devised that allows service providers to somehow specify trust anchors it 'knows' (e.g., specific persons within institutions) along with their representation in various web-of-trust networks, an approach that fits well if the service providers involved are provided by, or specific to, a virtual organization or collaboration.

3) LoA determination

A web-of-trust based authentication assurance is built from the accumulation of assertions of opinion/judgment by others. It is emergent or generative and is more a matter of judgment than fact. It is an establishment of reputation, as rendered by the attestation service based on a knowable and refutable set of attestations. For example, the trustworthiness that a user's identity is associated to an account is a construct of one or more judgments of other users about this association. Rarely do these sources agree, often because they base their judgment on varying data/experience. There is currently no clear agreement about how to convert the attestations into authentication LoAs. Likely parameters have been determined (number of attestations, the LoA of the helpers, etc.). Evaluation of the model and application in real-life settings has to turn out what suitable parameters are. Inspiration may be obtained from the work of Jøsang [10] and Neisse [11].

In the protocol description, we mentioned that the web-of-trust approach does not fit in the existing LoA frameworks defined by ISO/IEC 29115 and STORK QAA. These frameworks assume there is a central authority that issues the authentication solution and takes care of its binding to a user identity after some form of identity verification. In the web-of-trust based model, the verification role of this central authority becomes less important, i.e., this is done via claims of other users. Adoption of the web-of-trust model in these frameworks is one approach but could take a long time. Another approach is to register our web-of-trust based assurance profiles at the global IANA registry that has been setup for this purpose [12]. The registry is intended to be used as an aid to discovering LoA definitions in protocols that use a LoA concept, including Security Assertion Markup Language (SAML) 2.0 and OpenID Connect. The drawback of a registry approach is that it doesn't provide the registered LoA schemes with any formal status, i.e., it doesn't make them standards that are accepted on a global scale. On the other hand, conforming to standardized frameworks such as ISO/IEC 29115 or STORK QAA provides such a formal status and will make the attestation service more useful in a broader context.

4) Trustworthy exchange of vouching code

The approach implicitly assumes that the Helpers somehow identify and authenticate the Askers via physical

contact or another means that mediates physical communication like a mobile phone call or video session. It doesn't prevent the Asker to send an e-mail to the Helper with the vouching code. This weakness can be mitigated by explicitly asking the Helper to confirm that he had physical or mobile phone contact with the Asker for passing the vouching code. Another option would be to use a customized mobile app that facilitates the exchange of the vouching code to another mobile phone, i.e., the code is only exchanged if the mobile phones are shaken together. This option proves togetherness but excludes the use of remote communication channels such as the mobile phone or a video session. Consequently this narrows down the number of possibilities for exchanging the vouching code in a trustworthy manner.

5) Bootstrapping

Bootstrapping always remains an issue in web-of-trust approaches. The Attestation Service must have sufficient access to social networks or other webs of trust to reliable determine suitable Helpers. Though social networks and interfaces to them are readily available, they need to be made available to the Attestation Service. By making the Attestation Service part of an existing federation and by seducing users to link LinkedIn, Facebook or Google+ accounts to their federated account the bootstrapping problem can be tackled.

6) Usability

Usability is a potential weakness. Particularly in terms of comprehensibility: will the user understand why he/she has to login to the attestation service and pass vouching codes to helpers in order to increase the LoA of their authentication? Users may abort the vouching process because they do not understand why it is needed and consequently may lose confidence in the system. Lack of usability may come at the cost of adoption.

Also, some effort of the Helpers is required. However, Helpers will often have sufficient incentives to attest, e.g., because they need to collaborate with the Asker or want to share something that requires a high LoA. Since the assumption is that the Helpers in some way know and are connected to the Asker via one or more Webs of trust, allowing the Asker to include a reason for vouching in the request may provide further incentive for the Helpers to vouch for the Asker. For instance, the Asker needs to access a Virtual Organization database that is administered by the Helper. These incentives should cater for a reasonably quick enhancement of the user's authentication LoA.

C. Opportunities

1) Useful webs of trust are readily available

Existing webs of trust such as LinkedIn, Facebook, PGP or identity federations are readily available and their exploitation provides sufficient trustworthiness for authentication LoA enhancement purposes.

2) Attribute validation

Many commercial service providers offer discounts for e.g. students or members of a certain community. For these services it is critical to reliably validate the fact if a user is indeed a student or community member, as this is the basis for the discount provided. Other attributes are convenient, but could also be provided by the person directly. As the discounts for students and members are often considerable, these services are highly valuable for users. Attributes such as group membership and age are often used for authorization purposes and must be reliable too.

The Attestation Service can fulfil this need by acting as an attribute validation service. It can ask the Helpers to validate the attributes it has obtained from the Asker's identity provider. Additionally it can ask the Asker to self-assert several attributes (e.g. mobile phone number or gender) and ask the Helpers to validate the assertions. These Helper evaluations will increase the assurance level of the attribute. Similarly to authentication LoAs, this also introduces the need for attribute LoAs. Defining an attribute LoA framework is beyond the scope of this work. An initial attempt is made in the STORK2.0 project [13]. The attribute LoA solution allows the Attestation Service to provide the attributes during authentication, i.e., the service provider is informed about the assurance of the attribute.

Attributes such as student, mobile phone number, e-mail address, group membership and last name are likely to change in time. The reliability of the attestations made by helpers regarding these attributes is time-dependent and has to decrease in time. Consequently, the validation of attributes by helpers should be done one a frequent basis.

The identity providers in existing federations make explicit assertions about the user's identity, e.g., that he/she is a student at the University of Amsterdam. The attestations of other users easily fit into the "claims" architecture of the federated identity infrastructures, and service providers can readily judge the validity of a particular claim based on the authority ascribed to the identity provider in the context of a federated trust framework and the domain. For example, the University of Amsterdam identity provider is arguably definitive regarding the claim that the user is a student, but it is not authoritative for the student's financial status. A project manager is authoritative for the researcher's project membership and a government population register for the age of a student. So, for validation of attributes it is extremely important to know who is authoritative to do so. In a web-of-trust model this can be compensated by using large numbers of attestations: if a large number of helpers attest that a user is of a certain age then this will probably the case. Using large numbers of attestation may also result in large numbers of negative attestations. This may for instance be the case for the validation of membership of a small project team. Only the team members may give positive feedback, whereas the many more other helpers from outside the team may give negative feedback. The context should be taken into account to optimize the validation feedback from the web-of-trust.

D. Threats

1) Loss of privacy

The web-of-trust approach requires intensive linking social network accounts and mining of social network graphs. The Attestation Service potentially obtains insight in the social network of the user and of its connections. Without

 ISBN: 978-1-61208-427-5

proper security measures this may provide a huge privacy threat that will make users reluctant to use the system.
Alternatively, for those concerned that a third party may eventually abuse or be compelled to reveal the social network, decentralized secure computation could produce the aggregate values without a single party having access to the full social network, though such techniques incur substantial computational cost. NodeRank is a decentralized algorithm similar to PageRank that can assign reputations using a social network [14]. Alternately, one can propagate reputation ratings along the social network, where each agent receives information about potential targets through referral chains [15][16]. Cryptographic techniques can further improve decentralized algorithms by allowing precise control over the distribution of information among participants without requiring a trusted intermediary.

E. Summary

Most weakness can be mitigated by the opportunities and threats by strengths. However, two challenges remain to be addressed: usability and liability. The latter can be tackled by integrating the attestation service into the existing trust fabric of the federation (i.e., it becomes a federated service) and possibly by limiting (specific) attestations to a certain context (e.g., membership of a specific organization). The usability challenge strongly depends on how things are presented to the user. This will be the main aspect of the evaluation activity later on in the project.

Looking at web-of-trust LoA enhancement from a business perspective the following question immediately pops into mind: is there a business case for an attestation service? Since there is an increasing need for stronger authentication solutions and physical registration is costly, one would say so. Typically authentication solution service providers could benefit from an attestation service, particularly if standardized frameworks such as ISO/IEC 29115 adopt the approach. An additional value of the attestation service is the opportunity to use it for attribute validation by the web-of-trust. There also is an increasing need for reliable attributes, maybe even more than strong authentication. The sum of all digitally available information about an individual offers enormous potential value [17]. Applications leveraging personal data can boost efficiency, focus research and marketing, and spur the creation of personalized products and services. An important requirement is that the identity attributes are reliable. The attestation service has the ability to meet this requirement

VII. RELATED WORK

The idea of using a web-of-trust is not new and many other reputation systems involve the relationships of participants in the computation of the reputation. Models exist that combine transitive trust (as in certificates or PGP keys) with a reputation rating: If a participant A trusts participant B (with a certain rating) and participant B trusts participant C (with a certain rating), then participant A trusts participant C (with a rating as a function of the other two ratings) [18].

Another way to assign reputation based on social network structure considers each link in the network as an implicit recommendation for a person. Alternatively, weights can be added to the links by allowing users to privately rate their contacts based on characteristics such as trustworthiness. One can then apply a PageRank-like algorithm to assign reputations to individuals [19]. Because PageRank is based on the global structure of the network, it is more difficult to spoof than local network properties, as it is not sufficient to have just anyone recommend a user, but they need to have high reputation themselves.

Brondsema and Schamp have created a system called Konfidi that combines a trust network with the PGP Web-of-Trust [20]. The system implements a metric and mechanism for inferring the trust on the networks formed. The generated network creates trust pathways in between email sender and receiver that can be crawled and using trust mechanisms and metrics, trust values are inferred. This approach has to be extended with LoA-determination functionality to make it suitable for authentication LoA statements.

Calculating trust from social network aggregation is not new [21][22]. These approaches are solely based on the number of claims about a user and do not take into account other trust aspects such as the duration of the connection, presence of the connection in multiple social networks or overlapping attributes like skills and context (e.g. colleague, friend or group membership).

An interesting example is Lenddo [20]. Lenddo is an online platform that utilizes connections, relations and reputation from multiple social media sites such as Facebook to build a credit rating. At Lenddo, everything revolves around the LenddoScore. This number, ranging from 0 to 1,000, is a universal measurement of the user's trustworthiness, with 1,000 being the highest value. Using a proprietary and evolving algorithm, the rating is graphically plotted across categories like Social Data, Trusted Connections, and Financial Performance. This score is what helps the user to obtain approval for loans and services. Lenddo uses social data to ensure that the user is who he says he is. Lenddo also analyzes the user's connections and how strong they are; Lenddo only takes into account the strongest interactions. In many cases this means family, close friends, and coworkers.

These models focus primarily on the calculation of trust and reputation, whereas this work focusses on the translation of crowdsourced trust about an identity into authentication assurance.

VIII. CONCLUSION

Web-of-trust provides an interesting identity proving mechanism that can be used in registration for authentication to attain LoA 2 or 3. Ideally, it should be used in situations where the authentication means has a higher assurance level than its enrollment process (including identity registration and proofing). This could be due to the fact that physical registration was not possible or too expensive. For example, in case of an international collaboration where distance, language or poor electronic communication are barriers to proofing. The main issue of utilizing web-of-trust for

 ISBN: 978-1-61208-427-5

authentication purposes is related to usability, so it is advised to maximize usability in any implementation thereof.

Altogether, this means the applicability of web-of-trust authentication, with regard to necessary level of assurance, alternative registration processes and usability, is use case sensitive. For example, Facebook already has a functionality where users are asked to confirm a photo as their friend, since this concerns an easy extension for a social networking website. The concept of introducing your friends is intuitive, however its implementation for digital authentication is less straightforward.

Future work in the area of web-of-trust for identity management may consist of the following research activities:

- Further optimization of the algorithms and metrics for determining the authentication LoA based on claims from the web(s) of trust.
- Pilot studies to collect user feedback in order to evaluate the approach.
- Further optimization of the algorithms and metrics for determining suitable helper candidates from the web-of-trust. This activity involves complex data mining and analytics.
- Exploration of the use of web-of-trust for other identity-related aspects beyond authentication. Possible aspects are the use of web-of-trust for attribute validation (e.g., is the user indeed a student or older than 18 years?), for authorization purposes, or for linking different user accounts (e.g., the communication of a shared attribute that enables linking).

ACKNOWLEDGMENT

This work is sponsored by the Géant3plus Open Calls program.

REFERENCES

[1] ISO/IEC 29115:2013 Entity authentication assurance framework, available from www.iso.org.

[2] B. Hulsebosch, G. Lenzini, H Eertink, STORK Quality Authenticator Scheme, Deliverable D2.3, March 2009, available from www.eid-stork.eu.

[3] More information about PGP is available at www.pgpi.org.

[4] More information about the Friend of a Friend (FoaF) is available online at www.foaf-project.org.

[5] J. Brainard, A. Juels, R.L. Rivest, M. Szydlo, M. Yung, "Fourth Factor Authentication: Somebody You Know," in ACM CCS, 2006, USA, pp. 168–178, doi:10.1145/1180405.1180427.

[6] E. Carrara, G. Hogben, "Reputation-based Systems: a security analysis," ENISA position paper, October 2007.

[7] M. R. Henzinger, R. Motwani, C. Silverstein, "Challenges in web search engines," Newsletter ACM SIGIR Forum, Vol 36, Issue 2, Fall 2002, pp. 11-22.

[8] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," Science, vol. 298, pp. 824-827, 2002, doi:10.1126/science.298.5594.824.

[9] More information about Trulioo is available online at www.trulioo.com.

[10] A. Jøsang, R. Ismail, C. Boyd, "A survey of trust and reputation systems for online service provision," Decis. Support Syst., vol. 43, March 2007, pp. 618-644, doi:10.1016/j.dss.2005.05.019.

[11] R. Neisse, "Trust and privacy management support for context-aware service platforms," PhD thesis, University of Twente. CTIT Ph.D. Thesis Series No. 11-216 ISBN 978-90-365-3336-2, 2012.

[12] The LoA Registry, more information available online at http://levelofassurance.org/process.html.

[13] STORK2.0 project, more information is available online at www.eid-stork2.eu.

[14] P. Li, X. Qiu, NodeRank: An Algorithm to Assess State Enumeration Attack Graphs, 8th international conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp. 1-5, doi: 10.1109/WiCOM.2012.64785852012.

[15] B. Yu, M. P. Singh, "A social mechanism of reputation management in electronic communities," Proc. 4th International Workshop on Cooperative Information Agents IV, Springer-Verlag London, 2000, pp 154-165, ISBN:3-540-67703-8.

[16] G. Zacharia, A. Moukas, P. Maes, "Collaborative reputation mechanisms in electronic marketplaces," Proc. 32nd Hawaii Intl. Conf. on System Sciences (HICSS), 1999, vol. 8, p. 8026, ISBN:0-7695-0001-3.

[17] Boston Consultancy Group, The Value of our Digital Identity, 2012, available online at www.libertyglobal.com/PDF/public-policy/The-Value-of-Our-Digital-Identity.pdf.

[18] F. Kerschbaum, J. Haller, Y. Karabulut, P. Robinson, "PathTrust: A Trust-Based Reputation Service for Virtual Organization Formation," Proc. 4th Int. Conference on Trust Management (iTrust), vol. 3986 LNCS, 2006, pp. 193–205.

[19] L. Page, S. Brin, R. Motwani, T. Winograd, "The pagerank citation ranking: Bringing order to the web," Technical report, Stanford Digital Library Technologies Project, 1998.

[20] D. Brondsema, A. Schamp, "Konfidi: Trust Networks Using PGP and RDF," Proc. Of the WWW'06 Workshop on Models of Trust for the Web (MTW'06), Edinburgh, Scotland, UK, May 22, 2006.

[21] S. Noh, "Calculating trust using aggregation rules in social networks", Proc. 4th international conference on Autonomic and Trusted Computing, Hong Kong, China, pp 361-371, 2007, doi:10.1007/978-3-540-73547-2_38.

[22] H.R. Singh, A. Neelima, L.S. Singh, S.Ib. Singh, "A Model of Computing Trust in Web Based Social Network Using New Aggregation and Concatenation Operators," International Journal of Computer Science and Network, Volume 2, Issue 4, August 2013, IJCSN International Journal of Computer Science and Network, Volume 2, Issue 4, August 2013, ISSN:2277-5420.

[23] Lenddo, more information available online at www.lenddo.com/pages/faq.

 ISBN: 978-1-61208-427-5

Different Aproaches to Security Incidents and Proposal of Severity Assessment of Security Incident

Lukas Kralik, Roman Senkerik, Petr Stipek
Faculty of Applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
e-mail: kralik@fai.utb.cz

***Abstract*—This paper presents comprehensive theoretical background for future work, which will be aimed on multi-criterial evaluation and assessment of security incidents and proposal of methodology focused on audits of security incident management. This paper describes and comments three different points of view on security incident according to international standards or law (Cyber Security law in Czech Republic). The paper is mainly intended for Czech companies since it is based on project about Cyber Security Level in Czech Companies. Some criteria for assessment and evaluation of severity of security incident are proposed at the end of this contribution.**

Keywords-cyber security; security; incident; assessment of severity; ISMS; information security; incident management.

I. INTRODUCTION

An issue of security incidents and their resolving is inseparably connected with the field of ICT. It is necessary to look for more and more effective ways to prevent security incidents due to the increasing heterogeneity, complexity and pressure of confidence, integrity, availability or non-repudiation. It does not matter what it is monitored, it is important to be always prepared to act appropriately. Each security incident is bound with time pressure, which requires automated and clearly defined steps. A severity assessment of a security incident is absolutely necessary since it strongly affects a readiness for next incident.

This paper is divided into 4 sections which deal with theoretical aspects on the field of security incident. Section 2 describes basic terms and elements for resolving the security incident. Also there is a brief explanation of a security incident resolving in few steps. The next section is devoted to the definition of a security incident. This paper will serve as theoretical background for project about cyber security level in Czech companies. For that reason, there are 3 different definitions on the basis of used standards in Czech companies. The last section is the shortest but it is the most important. This section shows a conceptual proposal of severity assessment of the security incident. An objective of future work is to extend this proposal for multi-criterial evaluation. On the basis of this evaluation, it will be possible to assess occurred security incidents.

II. SECURITY INCIDENT – BASIC TERMS

A security incident is an event in information system, which caused disruption of confidence, integrity, availability or non-repudiation of information due to the failure of security measures or violation of security policy [1]-[5].

A suspected violation of a security policy or an attempt to overcome security measures is very often regarded for a security incident. A security incident usually has the following course: Incident Detection - Analysis of the Incident - Response to the Incident. Detection may be either automatic based on the information from some monitoring system, or manual, i.e., the incident is reported by someone. The company, which wants to deal with the security incidents and effectively solve them, should have an appropriate security standard and also it must properly present such standard to employees. The next step is formation of a team, which will be responsible for receiving reports, evidence and solving of incidents, etc. In many cases, this team is called Information Security Incident Response Team (ISIRT). The amount of ISIRT members depends on the total number and frequency of security incidents and of course on the size of company. For a proper function, ISIRT must have an adequate equipment, means and mainly authority [5][8]-[11].

The question is than as to how to determine the severity of the incident. There are many possible ways and approaches. The severity of the incident can be determined based on a value of an impact. In other words, the incident had financial or non-financial impact to the company. Another solution is to determine the severity of the incident according to the number and expertise of people who have to deal with the incident (more details are given in Section 3). It can be assumed that different number of people or teams with diverse levels of knowledge will participate in solution of various incidents [7]-[9].

A. Security standard

Each security standard must contain three basic elements. The first one is a definition of the security incident. The security incident must be clearly and understandable described with appropriate examples. These examples should be placed in attachment.

The next is information about security incident report. Contact should involve address on the intranet, e-mail, phone and office or workplace address because it is necessary to

 ISBN: 978-1-61208-427-5

take into account the simple fact that the network infrastructure may not work.

And the last one is a structure of a security incident report - form for reporting incidents [5][10].

B. Security incident log

Creation of security incident log is necessary for successful resolving of particular incident. Information listed in this log includes:

- When the incident has occurred - due to the fact that the incident may be related to other events, it is always advisable to ascertain the exact time.
- Where the incident has occurred - the exact place and its description will enable the investigative team to respond quickly.
- Who committed the incident - the identity of the intruder can sometimes be difficult to identify, but we should try to get about him as much relevant information as possible.
- How the incident has occurred - sometimes we do not have enough information, but we should try to build a probable scenario describing the incident.
- What was the target of an attack - we should also distinguish whether the system was directly attacked or used to preparation for another attack.
- Which security attribute was compromised - integrity, confidentiality, availability and/or non-repudiation.
- What was the nature of the incident - if the incident was intentional or unintentional. And if unintentional, thus if there was negligence or lack of knowledge of security policy.
- What measures have been overcome - whether the measures at the physical, logical, organizational, personnel or technical security.
- What asset has impaired - hardware, software (operating system, applications, databases), network, data, etc.
- What is a probability that the incident will be repeated again - rather low, medium, high or certain [5][10].

C. Equipment of ISIRT

The team should have developed procedures for dealing with specific types of incidents, and these procedures should be still updated with new types of incidents occurring. Also they should have prepared a communication plan to make it clear who has to inform whom, or who decide on further action etc.

A basic equipment of this team is a common room (war room), where it will be possible to meet and agree on the next steps in the event of an incident.

Last but not least, they need access to an adequate software and hardware resources - for example, the team will need to make a copy of configuration, logs or possibly an entire partition of the infected system.

D. Simplified procedure for investigation of incident

The whole procedure has 7 steps. The biggest problem in practice is in step 3. A top management usually requires immediate recovery of operations, thus there may be no time for ensuring clues and finding causes. However, ignoring this step makes environment/conditions for another step No. 6 more difficult. There should be proposed appropriate measures to prevent recurrence of incident. Choosing a suitable measure is so difficult, thus the company has no other option than hope that the incident will not occur again [3]

1. Identify where a security incident has occurred;
2. As quickly as possible prevent further damage;
3. Analyze cause of security incident and ensure clues for further analysis;
4. Remove a cause and restore functionality;
5. Assess damage;
6. Design and implement appropriate measures to prevent a recurrence of this incident;
7. Inform others (employees, top management...) with results of the investigation [2][8][9].

III. DEFINITION OF THE SECURITY INCIDENT

Many companies deal with incidents, but what kind of incident? Is it a computer, cyber or information incident? The usage of this term without any “additional specifying word/phrase” is quite problematic. There are 3 main and different points of view on definition of the security incident [1][3]-[5].

The definition of a security incident according to**:**

1. Cyber Security Law
2. NIST 800-61 (Computer Security Incident Handling Guide)
3. ISO/IEC 27001 - part of the growing ISO/IEC 27000 family of standards; information security management system (ISMS)

In Czech Republic, a new law about cyber security came in 1.1.2015. The main objective of this new law is to increase an overall security of cyberspace and set up a mechanism for active collaboration between business sector and public authorities. This implies new duties for companies. On these facts it was prepared this paper, whose main goal is to

 ISBN: 978-1-61208-427-5

provide basic overview about this field and mainly help with an implementation of an incident management in companies. In the implementation phase it is placed emphasis on a determining a severity of the security incident.

A. *Security incident according to Cyber Security Law*

In §8, section 2, the cyber security incident is defined: "*Cyber security incident is a cyber security event, which represents violation of information security in information systems or violation of security services and electronic communications networks*."[5]

For understanding of this definition it is necessary to look into section 1, as to how security event is defined. There is: "*Cyber security event is an event that can cause violation of information security in information systems or violation of security services and electronic communications networks.*" [5]

Information according to §2 d) means to ensure the confidentiality, integrity and availability of information. Cyberspace is defining as a digital environment enabling the creation, processing and exchange of information, consisting of information systems and services and electronic communications network. [5]

B. *Security Incident according to NIST 800-61*

A computer security incident is defined in NIST handbook 800-61 (Computer Security Incident Handling Guide, chapter 2.1) as: „A *computer security incident* is *a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.* " [1].

Even here, the term event appeared and it is specified as: "*An event is any observable occurrence in a system or network*." and "*Adverse events are events with a negative consequence…*" [1].

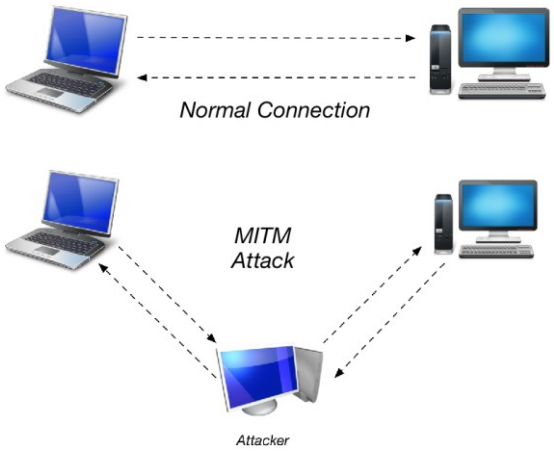


Figure 1. Principle of MITM attack

In the aforementioned handbook, it is possible to find attack vectors, which are removable media and e-mail (this way the malicious code can spread); Distributed Denial of Service (DDoS) attacks (Figure. 1), password guessing, finding vulnerabilities on web sites, **impersonation, spoofing**, Man-in-the-middle (MITM) attack (See Figure. 2), fake access points, violation organization's security policy, loss or stolen devices or media etc. [7][10][12].

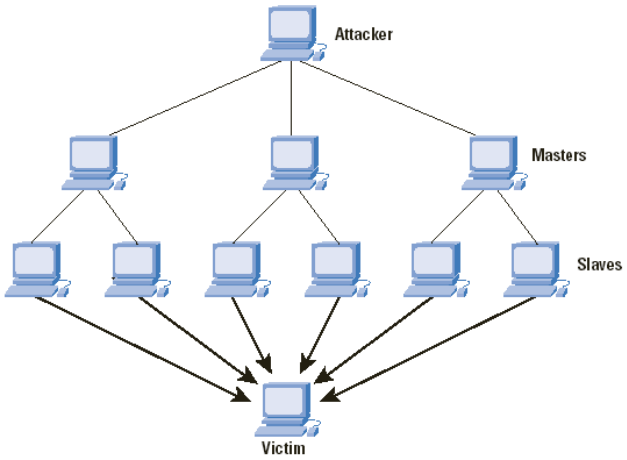


Figure 2. Principle of DDoS attack [6]

C. *Security Incident according to ISO/IEC 27001*

In chapter 3.6 of this standard, the security incident has following description: "One or more unwanted or unexpected security events for which there is a high probability of compromise of the organization's activities and threats to information security." Security event is defined in previous chapter 3.5. Security event is identifiable state of the system, service or network, pointing to a possible violation of security policy or failure of security measures. It may also be a different situation had not occurred before, which may be important in issue of information security [2]-[4].

D. *Comparison of definitions*

Each security incident should be caused by force majeure or violation of security policies. For the purpose of the project there were used 3 different definitions. Comparison between these definitions is described in following table (table 1) where:

"**?**" – the source of security incident is not clear if the definition is including it;

"**0**" – the source of security incident is not included in the definition;

"**1**" – the source of security incident is included in the definition.

 ISBN: 978-1-61208-427-5

TABLE I. SOURCES OF SECURITY INCIDENTS INCLUDED IN DEFINITION

	Force Majeure	Violation of Security Policies	
		intentionally	*unintentional*
Cyber security law	?	?	?
NIST 800-61	0	1	1
ISO/IEC 27001	1	1	1

The main standard for Czech companies is the cyber security law. The definition according to this law is quite wide, but it is not completely clear if cyber security incident represents also the violation of security by force majeure. Into this category it is classified for example interrupting of critical network infrastructure or servers due to flooding. Further, there should be a violation of security policy and standards that operator of critical infrastructure certainly published and it is mandatory for operator's employees.

The definition according to NIST 800-61 strictly claims that there has been a violation of security policies. Simultaneously there are no references to force majeure and the definition is clearly distanced from security violation due to natural disasters, blackout, etc.

Probably the best definition of security incident is presented in ISO/IEC 27001. It possible to notice that failure of security measures is also mentioned in addition to violation of security policies. Failure may be an unexpected event that could significantly compromise the security of information. However, it is questionable whether to be considered as a security incident, virus detected at the workstation (PC) and removed by antivirus, unplanned downtime of the system, utilization of an employer's mean for private purpose or retention and disposal of confidential documents on the table.

IV. ASSESSMENT OF SEVERITY OF INCIDENT

To correctly determine the severity of the incident it is very often a problem. In addition, the severity may vary throughout the life cycle of the incident. For example, at the beginning of the investigation of the incident, it may seem that this is a security incident with a negligible impact on the company and later on during the investigation it may prove that the original assumption was wrong.

If companies already have established a process that could be used with some exaggeration as an incident management and the severity of each incident is determined in this process, then their approach is very different. It is understandable that different companies use various number of degrees to reflect the severity of the incident and also individual levels have other names. However, it is striking that for determining the degree of severity, the companies do not have defined clear rules.

If company conducted a risk analysis then it can be relatively easy to determine the severity of incident based on the value of the asset whose confidentiality, integrity or availability has been or may be compromised. However, as already mentioned in section II, there are more possibilities. A proposal of criteria for determining the severity of the incident follows:

The severity should be defined by 4 levels:
- low
- middle
- high
- critical

Depending on the amount of affected users:
- one or few users
- whole department
- whole branch
- whole company

According to a level that will deal with the incident:
- technical (IT) support
- lower management
- middle management
- top management

Who should be familiar with the incident:
- one or a few employees of the company
- all employees
- own employees and persons outside the company
- own employees and the public

By level of expertise:
- first level of support
- system administrator
- security expert
- security company

Regardless of the size and scope of company, these four levels for assessment of the severity of the incident should be enough for most companies.

V. CONCLUSION

Security incidents and their solutions are an essential part of life of IS/ICT manager as well as of ordinary users. Absolute security of an information system is not guaranteed by implementation of any security policy. Although the implementation of various security functions and measures are part of ensuring security, vulnerabilities remain in the information system and these vulnerabilities represent risks. The existence of these vulnerabilities is the possibility of the security incident with direct or indirect impact on everyday operations of companies. Therefore, it is essential that each company pay attention to the definition and the implementation of security management system, its control and audit. At the same time companies should also deal with efficient and professional management of security

 ISBN: 978-1-61208-427-5

incidents. Incidents can be controlled intuitively or in structured way - professionally. Only a professional approach allows gaining benefits from security incidents - experience, skills and knowledge from solutions of previous security incidents.

ACKNOWLEDGMENT

This work was supported by grant No. IGA/FAI/2015/039 and IGA/CebiaTech/2015/036 from Internal Grant Agency of Thomas Bata University in Zlin; further by financial support of research project NPU I No. MSMT-7778/2014 by the Ministry of Education of the Czech Republic and also by the European Regional Development Fund under the Project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

REFERENCES

[1] NIST, "Special Publication 800-61 – Computer Security Incident Handling Guide, Revision 2: 800-861", 2012.

[2] International Organization for Standardization ISO/IEC 27000-Information technology-Security techniques-Information security management systems-Overview and vocabulary.

[3] International Organization for Standardization ISO/IEC TR 18044:2004- Information technology - Security techniques - Information security incident management.

[4] International Organization for Standardization ISO/IEC 27001 - Technology-Security Techniques - Information Security Management Systems-Requirements.

[5] Czech. Law nr. 181/2014 sb. Cyber Security Law. 2014.

[6] P. Doucek "IS/ICT Security Incidents and theire Solutions," System Integration vol. 3, Prague 2005, pp. 77-85.

[7] C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed Denial of Service Attacks," In: Internet Protocol Journal [online]. Volume 7, Number 4, San Jose, USA: Cisco Systems, Inc, 2004, ISSN 1944-1134, online: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html, [retrieved: July, 2015].

[8] L. Lukas, M. Cahlik, and L. Kralik, "Protection of Data Centers – Physical Protection, "Recent Advances in Information Science, Proceedings of the 3rd European Conference of Computer Science (ECCS '12). Paris, France WSEAS Press, 2012, 171-176. ISBN 978-1-61804-140-1, ISSN 1790-5109

[9] L. Wan-Soo and J. Sang-Soo, "A Study on Information Management Model for Small and Medium Enterprises," Recent Advances in E-Activities, Information Security and Privacy, Spain, WSEAS Press, 2009, ISSN: 1790-5117. ISBN: 978-960-474-143-4

[10] K. Prislan and I. Bernik, "Risk Management with ISO 27000 Standards in Information Security," In Advances in E-Activities, Information Security and Privacy, Venezuela WSEAS Press 2010, ISBN: 978-960-474-258-5

[11] L. Kralik and R. Senkerik, "Proposal for Security Management System," Recent Advances in Electrical Engineering and Educational Technologies. Proceedings of the 2nd International Conference on Systems, Control and Informatics (SCI 2014), Athens, 2014. p. 77-80. ISBN 978-1-61804-254-5

[12] S. Fenz and A. Ekelhart. "Formalizing Information Security Knowledge" Book Formalizing Information Security Knowledge' (ACM, 2009, Edn.): 183-194

 ISBN: 978-1-61208-427-5