Ali Ismail Awad
Aboul Ella Hassanien
Kensuke Baba (Eds.)

Communications in Computer and Information Science 381

Advances in Security of Information and Communication Networks

First International Conference, SecNet 2013
Cairo, Egypt, September 2013
Proceedings

Springer

Communications in Computer and Information Science 381

Editorial Board

Ali Ismail Awad Aboul Ella Hassanien
Kensuke Baba (Eds.)

Advances in Security of Information and Communication Networks

First International Conference, SecNet 2013
Cairo, Egypt, September 3-5, 2013
Proceedings

Springer

Volume Editors

Ali Ismail Awad
Al Azhar University
Faculty of Engineering
Qena, Egypt
E-mail: aawad@ieee.org

Aboul Ella Hassanien
Cairo University
Department of Information Technology
Cairo, Giza, Egypt
E-mail: aboitcairo@fci-cu.edu.eg

Kensuke Baba
Kyushu University, Library
Fukuoka, Japan
E-mail: baba@soc.ait.kyushu-u.ac.jp

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-40596-9 e-ISBN 978-3-642-40597-6
DOI 10.1007/978-3-642-40597-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013946094

CR Subject Classification (1998): K.6.5, C.2.0, H.2.7-8, I.2.6, D.4.6, K.4.4

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Owing to its wide diversity of applications, information security is subject to intensive research by governmental and private institutes. The First International Conference on Advances in Security of Information and Communication Networks (SecNet 2013) was held at Cairo University, Cairo city, Egypt, during September 3–5, 2013. The goal of the conference is to bring together, in a friendly atmosphere, researchers and practitioners from academia and industry, and to provide a discussion forum for the sharing of knowledge and experiences.

The conference received 62 submissions in all areas of information and communication networks security from different countries such as the USA, Spain, UK, France, Australia, Canada, India, Kuwait, Malaysia, and Egypt. The conference Program Committee includes experts and recognized researchers from many countries including the UK, USA, Japan, Malaysia, India, Czech Republic, Italy, Taiwan, and Egypt. The worldwide participation in SecNet 2013 gave it a truly international scope. All submissions were reviewed by at least two independent Program Committee members. In all, 21 papers were accepted, with a total acceptance rate of 33.8%. The authors of accepted papers are thanked for revising their papers according to the suggestions of the reviewers. The revised versions were not checked again by the Program Committee, and therefore the authors bear full responsibility for their content.

This volume represents the revised versions of the 21 papers accepted for oral presentation, and it is organized into four main sections. The first section is titled "Networking Security", and it includes six papers. The second section is reserved for documenting the general trends in security, "Data and Information Security", and it includes five papers. The third section documents the research papers related to data authentication and user privacy, titled "Authentication and Privacy", and it comprises five papers. Finally, the fourth section is titled "Applications", and it includes five contributions related to the applications of information security.

The editors are indebted to the efforts of the Program Committee members in reviewing and discussing the papers. Springer's new Online Conference Service (OCS) provided great help during the submission, the reviewing, and the editing phases of the conference proceedings, and the editors are very grateful to the OCS staff for their help. As editors, we are very thankful to Alfred Hofmann and the excellent *Communications in Computer and Information Science* (CCIS) team at Springer for their support and cooperation in publishing the proceedings as a volume in the CCIS series. The editors would like to acknowledge the Scientific Research Group in Egypt (SRGE)

as the technical sponsor of SecNet 2013. Finally, the editors are thankful to the Organizing Committee and the members of SRGE for their volunteer work during the activities of the conference.

June 2013

Ali Ismail Awad
Aboul Ella Hassanien
Kensuke Baba

Organization

General Chair

Aboul Ella Hassanien, Egypt

Program Chairs

Ali Ismail Awad, Egypt
Kensuke Baba, Japan

Publicity Chairs

Ahmad Taher Azar, Egypt
Nashwa El Bendary, Egypt

Local Organizing Committee

Neveen Ghali, Egypt
Nashwa El-Bendary, Egypt
Mostafa Salama, Egypt
Mohamed Mostafa, Egypt
Heba Eid, Egypt
Kareem Kamal, Egypt
Mohamed Tahoun, Egypt

International Program Committee

Adel Alimi, Tunisia
Azizah Abd Manaf, Malaysia
Craig Valli, Australia
Dipankar Dasgupta, USA
Dusan Husek, Czech Republic
Ehab Mahmoud Mohammed, Egypt
Elsayed Mohamed, Egypt
Emilio Corchado, Spain
Eyas El-Qawasmeh, Kingdom of Saudi Arabia
Francesco Marcellon, Italy
Hala S. Own, Kuwait
He Debiao, China
Hideyuki Takag, Japan
Jude Hemanth, India
Kazumi Nakamatsu, Japan
Kensuke Baba, Japan
Lamiaa Ebakrawy, Egypt
Mahmoud Hassaballah, Egypt
Mohamed Hassan Essai, Egypt
Muhammad Younas, UK
Nashwa El-Bendary, Egypt

Neil Y. Yen, Japan
Omar F. El-Gayar, USA
Ravi Sandhu, USA
Salwani Mohd. Daud, Malaysia
Samy El-Ghoniemy, Egypt
Saru Kumari, India
Shampa Chakraverty, India
Shi-Jinn Horng, Taiwan
Soumya Banerjee, India
Tai-hoon Kim, Australia
Vaclav Snasel, Czech Republic
Waheedah Al Mayyan, UK

Table of Contents

Networking Security

Data and Information Security

Authentication and Privacy

Security Applications

NETA: Evaluating the Effects of NETwork Attacks. MANETs as a Case Study

Leovigildo Sánchez-Casado, Rafael Alejandro Rodríguez-Gómez, Roberto Magán-Carrión, and Gabriel Maciá-Fernández

Dpt. Signal Theory, Telematic and Communications, CITIC, Univ. of Granada
c/ Periodista Daniel Saucedo Aranda s/n, 18071, Granada, Spain
{sancale,rodgom,rmagan,gmacia}@ugr.es

Abstract. This work introduces NETA, a novel framework for the simulation of communication networks attacks. It is built on top of the INET framework and the OMNET++ simulator, using the generally accepted implementations of many different protocols, as well as models for mobility, battery consumption, channel errors, etc. NETA is intended to become an useful framework for researchers focused on the network security field. Its flexible design is appropriate for the implementation and evaluation of many types of attacks, doing it accurate for the benchmarking of current defense solutions under same testing conditions or for the development of new defense techniques. As a proof of concept, three different attacks have been implemented in NETA. The capabilities of NETA are exhibited by evaluating the performance of the three implemented attacks under different MANET deployments.

Keywords: Network simulation, network attacks.

1 Introduction

Network security is currently becoming one of the main problems for the development of new technologies and services in telecommunication networks. Hackers are constantly evolving towards new attack techniques and new target technologies at a very high speed [1] [2], thus making the task of building defense mechanisms a hard mission.

In this context, many efforts have been done by the research community to develop security defenses aimed at defeating attacks. The cycle is almost always the same: whenever a new attack technique or vulnerability is discovered by a researcher, a proof of concept implementation is built as a proprietary development, an evaluation of the capabilities of this technique done, and the development of effective defense techniques proposed.

As a result of this research methodology, although many researchers contribute their network attacks code, there is a lack of accepted implementations for the attacks that would allow to benchmark solutions against them.

Thus, it is desirable to have a common framework that would allow the development of implementations of network attacks and their defenses. This framework should allow to combine the execution of all the implemented attacks, in a

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 1–10, 2013.

similar way as hackers do, and also allow to test them on multiple technologies, protocols and scenarios.

In this paper we introduce a framework that we have developed and contributed to the research community, trying to fulfil the above conditions. NETA [3] (NETwork Attacks) is an OMNeT++ based network attacks framework, intended to provide a base reference framework to unify the attack development and simulation. NETA is extensible and offer a high degree of versatility for the development of new and heterogeneous network attacks. It aims at saving efforts in the attack development process employed for testing purposes, thus offering a useful tool for the research community in the network security field. NETA is publicly available for download in `http://nesg.ugr.es/index.php/en/neta`

The rest of the paper is organized as follows. Section 2 provides some related work regarding simulators and other similar approaches. The general architecture of the framework is presented in Section 3, where the main components and the design rules are explained. In Section 4, we describe the implemented attacks in this first release of the framework. Section 5 describes the experimental environment to test the framework, as well as the results obtained. Finally, conclusions and future work are presented in Section 6.

2 Related Work

Simulation is normally used to test network protocols and complex systems, offering the research community a good compromise between cost and complexity [4]. Nevertheless, the choice of the best simulator is not an easy task. It requires a previous study considering advantages and drawbacks.

According to [5] and [6] the simulators most widely used in the field of networking are: (*i*) Optimized Network Engineering Tools, *OPNET*, (*ii*) Network Simulator 2, *NS2*, and (*iii*) *OMNeT++*. They are all powerful discrete-event simulators for heterogeneous networks. It is remarkable the capacity of OPNET to execute and manage concurrently several scenarios and the rich set of protocols provided by NS2. Nowadays, OMNeT++ is becoming one of the most used ones due to the huge amount of frameworks (INET, MIXIM, etc) it offers, its higher flexibility, and its user-friendly GUI, among other advantages.

With regard to the simulation and the design of networks attacks, authors usually implement specific attacks by themselves with the aim of testing security proposals (detection or response-based), protocols performance and so on [7]. These attack implementations used to be private and, therefore, two different defense proposals can not be compared with the same attack implementation, making this comparison less accurate and reliable.

The authors in [8] provide an OMNeT++ based framework to simulate traffic patterns and DoS attacks over IP networks. However, they only implement a specific type of attacks and this framework is not extensible to implement other attack types. An attack simulation framework applied to WSNs is proposed in [9]. The authors present a procedure to simulate attacks by devising a particular attack language which describes the attack behavior. The framework

is extensible but it is not publicly available and it can not be applied to other environments different from WSNs. For these reasons, there is still a need for a general, extensible and versatile attack framework to be devised in order to address the previous drawbacks. NETA framework is proposed here as a solution.

3 NETA: A Simulation Framework for NETwork Attacks

We have built NETA as an OMNeT++ simulator framework built on top of the INET framework. NETA is intended to be widely used by the research community, considering that OMNeT++ is one of the most common simulation tools in the networking field. Additionally, NETA framework is based on the same idea as OMNeT++, *i.e.*, modules that communicate by message passing.

The general idea is to develop models in OMNET++ implemented as new nodes which can strike attacks, *attacker nodes*. In order to do this, the attacks are managed by the so-called *attack controllers*. These controllers manage one or more modules of a NETA framework attack node by sending *control messages*. These messages are sent from attack controllers to specific modules that implement a modified behavior for the attack. They are called *hacked modules* hereafter. For implementing this modified behavior, these hacked modules are inherited or replicated from INET modules and conveniently modified to obey the orders of attack controllers.

The design principles of the present framework follow two main rules:

Rule 1 *Any base framework we use must not be modified, e.g., when using INET modules, they should remain as the original one.*

This rule is intended to facilitate the compatibility with future releases of INET and other implementations. To accomplish this rule we just import the last version of INET framework and we do not carry out any modification on it.

Rule 2 *To modify the least possible the original code of the hacked modules.*

Obviously, in order to implement the desired attacks, it is necessary to modify the behaviour of the modules that will become hacked modules. However, this rule is intended to minimize these modifications as much as possible.

The creation of an attacker node can be summarized as: (*i*) add to the associated `.ned` file the controllers related to the attacks to be executed, (*ii*) create the associated control messages and, (*iii*) substitute the modules needed by these attack controllers for corresponding hacked modules.

Fig. 1shows the differences between a normal and an attacker node. The normal node is composed of simple and compound modules communicating among them. The attacker node is composed of the same number of modules but now controller modules are added. In addition, some of the modules are replaced by hacked modules, in order to allow the execution of attack behaviours when triggered by attack controllers.

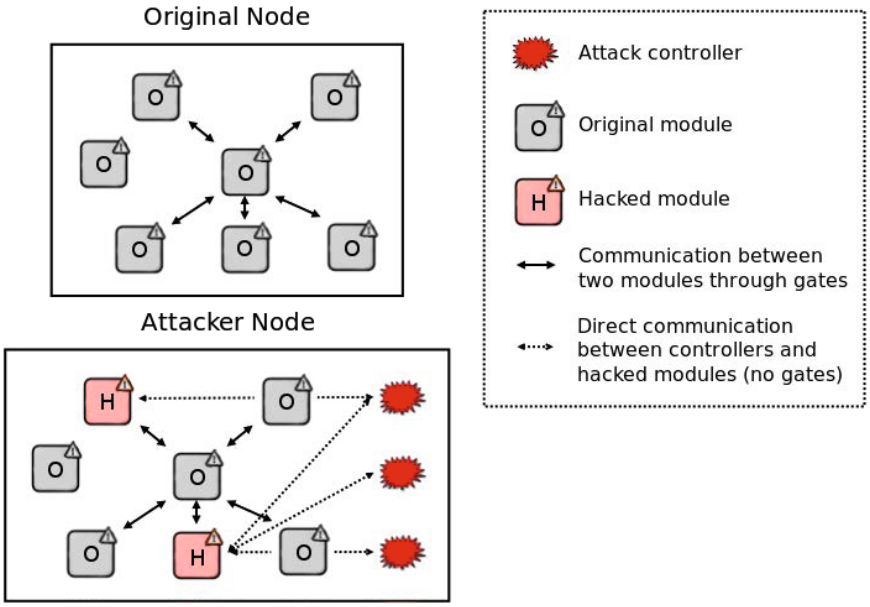


Fig. 1. Scheme comparison between an original node and its attacker in NETA framework

3.1 NETA Architecture

In the following we describe the main components of an attack in our framework: (*i*) attack controllers, (*ii*) control messages, and (*iii*) hacked modules.

Attack Controllers: modules which control the execution of the attack. They have the following properties:

- `attackType`: name intended to differentiate an attack to the rest of them.
- `active`: it indicates whether the attack is active in the simulation or not.
- `startTime`: the time at which the attack starts in the simulation.
- `endTime`: the time at which the attack ceases.
- `Attack specific parameters`: different configuration parameters depending on the specific attack functionalities.

The processes carried out by an attack controller for attack A_i in an attacker node can be summarized as:

1. To obtain the different hacked modules involved in the execution of attack A_i.
2. To activate those hacked modules in the attack node by sending, at start time, activation messages which can contain configuration information.
3. To deactivate the hacked modules in the attack node by sending a deactivation message at end time.

Control Messages: they are sent from attack controllers to the hacked modules involved in the attack execution. They transmit the information necessary for the activation and deactivation of the attacks. Additionally, these messages contain configuration information needed for the execution of the attacks.

It is important to remark that control messages are sent directly to a hacked module. This is the best option to accomplish the rule 2 of our design principles: "To minimize the modifications to the original code of hacked modules".

Hacked Modules: these are the modules whose behavior is modified in order to strike an attack. For example, a packet dropping attack usually requires a modification in the module that makes IP forwarding. Therefore, the implementation of a dropping attack implies the modification of the NETA IPv4 module, which behaves as a hacked module.

Note that there exists only one hacked module per modified module, and not a hacked module for every attack implementation. If two different attacks need to modify the same module, there will only exist one hacked module for them. For instance, as it will be shown, both delay and dropping attacks are related to the IPv4 module. Thus, a single hacked IPv4 module is needed for the implementation of the two attacks. This design is aimed to improve the flexibility of the framework, allowing the execution of more than one attack simultaneously, *e.g.*, delay and dropping attacks can be triggered by the same node only by including their attack controllers.

4 Implemented Attacks

This section exposes the attacks implemented as a proof of concept for the NETA framework. In the subsequent sections, for every implemented attack we describe: (*i*) the behavior of the attack, and (*ii*) the parameters which can be modified to configure the attack.

4.1 IP Dropping Attack

In the IP dropping attack, nodes exhibiting this behavior intentionally drop, with a certain probability, received IP data packets instead of forwarding them, disrupting the normal network operation. Depending on the application, it can turn the network much slower due to the existence of retransmissions, make the nodes waste much more energy resources, etc. The main parameter of our implementation of the dropping attack is:

- `droppingAttackProbability`: the probability of dropping a packet, defined between 0 and 1. By default, it is set to 0 which makes the attacker node to behave normally (no dropping at all).

4.2 IP Delay Attack

In this attack, a malicious node delays IP data packets for a certain amount of time. This can affect different QoS parameters (end-to-end delay, jitter, etc.), resulting in a poor network performance. The list of parameters in our implementation of the delay attack is:

- `delayAttackProbability`: the probability of delaying a data packet, defined between 0 and 1. By default, it is set to 0 which implies a normal behavior for the attacker node (no extra delay for any packet).
- `delayAttackValue`: the specific delay time applied to the packet. Note that this parameter could be specified by a statistical distribution. For this reason, it is defined as volatile, *i.e.*, it is modified every time it is accessed. By default, it follows a normal distribution with mean 1 second and standard deviation of 0.1 seconds.

4.3 Sinkhole Attack

In a sinkhole attack, a malicious node sends fake routing information, claiming that it has an optimum route and causing other nodes to route data packets through itself. Here, the attacker forge routing replies (`RREP`) to attract traffic. The list of parameters of sinkhole attack is:

- `sinkholeAttackProbability`: the probability of answering a `RREQ` message with a fake route reply (`RREP`), defined between 0 and 1. By default it is set to 0 which implies the normal behavior of AODV protocol.
- `sinkOnlyWhenRouteInTable`: if set to *true*, the sinkhole only sends fake `RREP` to requests for those the attacker node has a valid route, *i.e.*, routes existing in its routing table. Otherwise (false value), the node sends fake `RREP` to any `RREQ` message arriving, even if it does not know a valid route.
- `seqnoAdded`: the fake sequence number generated by the attacker node. It is added to the sequence number observed in the request. It can be different each time, if it is specified as an statistical distribution. By default, it follows a uniform distribution with values between 20 and 30.
- `numHops`: the fake number of hops returned by the attacker. By default, it is set to 1, indicating that the attacker reaches the end of the communication in only one hop.

5 Experimental Evaluation

In this section the experimental environment used to evaluate the aforementioned attacks is presented. Additionally, several tests have been made to verify the proper performance of every implemented attack, measuring its impact on the network according to different metrics.

Our aim here is to show the capabilities of the simulation framework, able to ease the work of extracting information about the attacks performance.

5.1 Common Experimental Environment

As a case study, a series of MANET deployments are simulated. The common parameters to all scenarios are described in what follows.

The simulation area is restricted to a 1000m x 1000m square, with each node having a communication range of 250m. The simulation time is set to 300s. The results have been derived by averaging (with different seeds) 50 simulation runs.

AODV and 802.11g are chosen as routing and medium access control (MAC) layer protocols respectively and the RTS/CTS mechanism is used to send packets. This last assumption is coherent with the mobility of nodes, as the lack of virtual carrier detection in such a mobility scenarios would imply a high number of collisions due to the hidden station problem.

The total number of nodes is 25, while the number of attackers varies from 1 to 3. The attacks are performed during the whole simulation time, and the corresponding *attack rate* is set to 100% where the *attack rate* is the probability of an attacker node to trigger its attack.

The number of application traffic flows is fixed to 21. Each flow performs as a Constant Bitrate (CBR) connection of 4 packets/s, where packet payload size is 512 bytes. The flows randomly start between 0.5 and 1.5 s and they end between 290 and 295 s.

We use a Random Waypoint Model (RWP) to simulate the movements of the nodes. The minimum speed is set to 1 m/s and the maximum varies from 5 to 20 m/s, with a pause time of 15 s.

5.2 Dropping Attack Evaluation

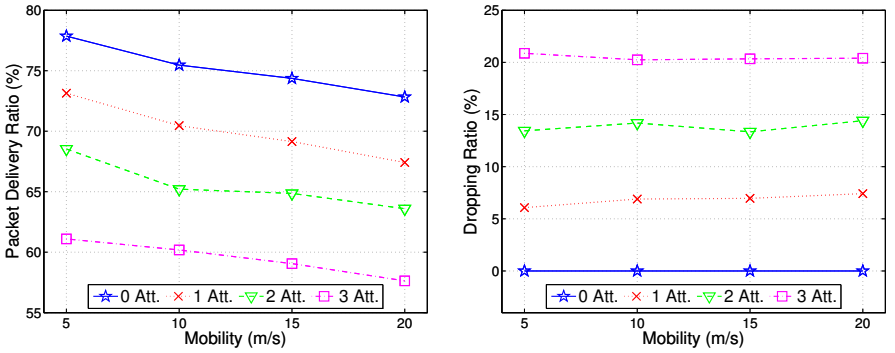
To evaluate the right operation of the dropping attack, the following performance metrics are defined:

- **Packet Delivery Ratio, *PDR* (%)**: total number of delivered data packets divided by the total number of transmitted data packets.
- **Dropping Ratio, *DR* (%)**: total number of data packets lost due to the execution of the attack divided by the total number of transmitted data packets.

As we can see in Fig. 2, if the number of attackers is increased, the PDR is deteriorated and the DR rises up. Additionally, the PDR decreases with the mobility, whereas the DR remains nearly constant. This is due to the fact that the mobility increases the number of packets lost by collisions and channel errors, while the number of packets lost as a consequence of the dropping attack remains constant.

5.3 Delay Attack Evaluation

The following performance metric is used to evaluate right operation of the delay attack:

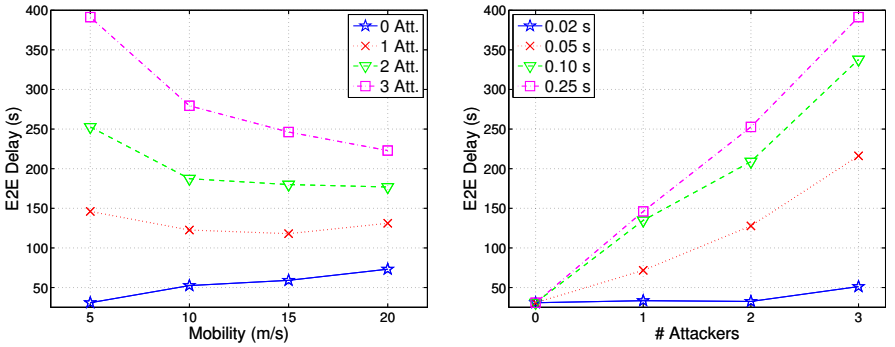


(a) PDR (b) DR

Fig. 2. *PDR* and *DR* as a function of the mobility speed and the number of attackers

– **End-to-End Delay, *E2ED* (s)**: the mean time employed by a data packet from its transmission until it reaches the destination. It is computed as the average of the specific E2ED of every packet in every flow, thus extracting the average E2ED for the whole network.

Here we have tested the delay attack as a function of (*i*) the number of attackers (Fig. 3(a)), and (*ii*) the delay used by the attackers (Fig. 3(b)). In the first case we fix the inserted delay to 0.25 s, and in the second one the mobility is set to 5 m/s. As expected, the average delay increases with the number of attackers as well as with the delay used by attackers.



(a) (b)

Fig. 3. *E2ED* for (a) different mobility speeds and number of attackers, with delay equal to 0.25 s and (b) different values of delay with a mobility of 5m/s

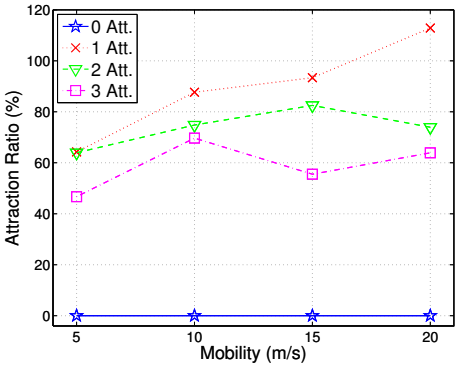


Fig. 4. *AR* for different mobility speeds and number of attackers

5.4 Sinkhole Attack Evaluation

To characterize the performance of sinkhole nodes we define the following metric:

– **Attraction Ratio, *AR* (%)**: the growth rate between the average number of packets received by sinkhole nodes and the average number of packets received by legitimate nodes.AR is computed as:

$$AR = \frac{\frac{1}{N_S}\sum_{i=1}^{N_S} pckt_i - \frac{1}{N_L}\sum_{j=1}^{N_L} pckt_j}{\frac{1}{N_S}\sum_{i=1}^{N_S} pckt_i} \cdot 100 \quad (1)$$

where N_S and N_L are the number of sinkhole and legitimate nodes respectively and $pckt_i$ the total number of packets received by the node i.

Fig. 4 shows how sinkhole nodes are attracting more traffic than normal nodes. Besides, we can see that AR decreases while the number of attackers increases. This is due to the fact that attackers compete between them to attract traffic, resulting in a lower AR. However, the total number of packets attracted by all the sinkhole nodes grows with the number of attackers.

6 Future Work and Conclusions

In this work, we have proposed NETA, a novel framework for the simulation of network attacks which has been built on top of the INET framework and OMNeT++ simulator.

NETA is composed of three main components: *attacks controllers* which manage the attacks execution, *hacked modules* which implement the actual behavior of the attack, and *control messages* which transmit the activation/deactivation information as well as configuration information from the attack controllers to the hacked modules. Moreover, three different attacks have been implemented as a proof of concept.

As a case study, we have considered realistic application scenarios by analyzing a series of MANET deployments. As shown, experimental results obtained prove the proper behavior of the implemented attacks. Additionally, we have slightly evaluated how the attacks affect the normal network operation.

This framework still need some improvements which are planned to be afforded in a near future. Specifically, we focus on implementing new and more complex attacks. We are also working on the development of different performance metrics which can be accurately used for benchmarking defense solutions as well as performance analysis under the same conditions.

Acknowledgment. This work has been partially supported by Spanish MICINN (Ministerio de Ciencia e Innovación) through project TEC2011-22579.

References

1. Jhaveri, R.H., Patel, S.J., Jinwala, D.C.: Dos attacks in mobile ad hoc networks: A survey. In: Proceedings of the 2012 2nd International Conference on Advanced Computing & Communication Technologies, ACCT, pp. 535–541. IEEE Computer Society (January 2012)
2. Yu, Y., Li, K., Zhou, W., Li, P.: Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. J. Netw. Comput. Appl. 35(3), 867–880 (2012)
3. Network Engineering Security Group (NESG): NETA: NETwork Attacks Framework for OMNeT++, `http://nesg.ugr.es/index.php/en/neta` (accessed April 25, 2013)
4. Lessmann, J., Janacik, P., Lachev, L., Orfanus, D.: Comparative study of wireless network simulators. In: 7th International Conference on Networking, ICN, pp. 517–523. IEEE Computer Society (April 2008)
5. ur Rehman Khan, A., Bilal, S.M., Othman, M.: A performance comparison of open source network simulators for wireless networks. In: IEEE International Conference on Control System, Computing and Engineering, ICCSCE, pp. 34–38. IEEE Computer Society (November 2012)
6. Kumar, A., Kaushik, S., Sharma, R., Raj, P.: Simulators for wireless networks: A comparative study. In: International Conference on Computing Sciences, ICCS, pp. 338–342. IEEE Computer Society (September 2012)
7. Ehsan, H., Khan, F.: Malicious AODV: implementation and analysis of routing attacks in MANETs. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, pp. 1181–1187. IEEE Computer Society (June 2012)
8. Gamer, T., Scharf, M.: Realistic simulation environments for IP-based networks. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. SIMUTools, pp. 83:1–83:7. ACM (March 2008)
9. Dini, G., Tiloca, M.: ASF: an attack simulation framework for wireless sensor networks. In: IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, pp. 203–210. IEEE Computer Society (October 2012)

Clustering Based Group Key Management for MANET

Ayman El-Sayed

Department of Computer Science and Engineering, Faculty of Electronic Engineering,
Menoufiya University, Menouf 32952, Egypt
ayman.elsayed@el-eng.menofia.edu.eg

Abstract. The migration from wired network to wireless network has been a global trend in the past few decades. The mobility and scalability brought by wireless network made it possible in many applications. Among all the contemporary wireless networks, Mobile Ad hoc Network (MANET) is one of the most important and unique applications. MANET is a collection of autonomous nodes or terminals which communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Due to the nature of unreliable wireless medium data transfer is a major problem in MANET and it lacks security and reliability of data. A Key management is vital part of security. This issue is even bigger in wireless network compared to wired network. The distribution of keys in an authenticated manner is a difficult task in MANET and when a member leaves or joins it need to generate a new key to maintain forward and backward secrecy. In this paper, we propose a Clustering based Group Key Management scheme (CGK) that is a simple, efficient and scalable Group Key management for MANETs and different other schemes are classified. Group members compute the group key in a distributed manner.

Keywords: Group Key management, Mobile Ad hoc network, MANET security, Unicast/Multicast protocols in MANET.

1 Introduction

Mobile Ad Hoc Network (MANET) [1, 2] is kind of mobile, multiple hops, and self-discipline system, not depend on the fixed communication facilities. Ad Hoc network is a series of nodes in structure which move anywhere at will, the network nodes distribute dynamically, nodes contact others through wireless network, every network node has the double functions as terminal and routers, the nodes are peer-to-peer, communicate with a high degree of coordination. Wireless Ad Hoc network is flexibility with a wide foreground of application [3]. A communication session is achieved either through single-hop transmission if the recipient is within the transmission range of the source node, or by relaying through intermediate nodes otherwise. For this reason, MANETs are also called multi-hop packet radio network [4, 5]. However, group key management for large and dynamic groups in MANETs is difficult problem because of the requirement of scalability, security under the restrictions of nodes' available resources and unpredictable mobility [6]. But the group key

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 11–26, 2013.

management protocols dedicated to operate in wired networks are not suited to MANET, because of the characteristics and the challenges of such environments [7]. So many researchers are interesting of group key management for MANET. In our issue, group key management means that multiple parties need to create a common secret to be used to exchange information securely. Without central trusted entity, two people that have not previously a common share key can create a key based on the Diffie-Hellman (DH) protocol [8]. By combining one's private key and the other party's public key, both parties can compute the same shared secret number. This number can then be converted into cryptographic keying material. It is called 2-party DH protocol that can be extended to a generalized version of n-party DH. In [9], the authors integrated the DH key exchange into the Digital Signature Algorithm (DSA) and in [10], the authors fix this integration protocols so that both forward secrecy and key freshness can be guaranteed, while preserving the basic essence of the original protocols. However, robust key management services are central to ensuring privacy protection in wireless ad hoc network settings. Existing approaches to key management, which often rely on trusted, centralized entities, are not well–suited for the highly dynamic, spontaneous nature of ad hoc networks. So many researchers are interesting to make proposals for key management techniques that are surveyed in [11] to find an efficient key management for secure and reliable. This paper proposes one of the key management schemes namely a Clustering based Group Key Management scheme (CGK) that is a simple, efficient and scalable Group Key management for MANETs. Group members compute the group key in a distributed manner. This hierarchical contains two levels only, first level for all coordinators of the clusters as a main group's members; it is called cluster head (CH), the second level for the members in a cluster with its CH. Then there are two secret keys obtained in a distributed manner, the first key among all the CHs and the second key among cluster's members and its CH. CGK uses double trees in each cluster for robustness and avoid fault tolerance. Also group key management is to ensure scalable and efficient key delivery, taking into account the node mobility. The remainder of this paper is organized as follows: Section 2 reviews related work such that MANET routing protocols for both unicast and multicast and security requirements. Also this section describes the overview of MANET key management and short note about our proposal. Details of our group key management scheme are described in Section 3 and our scheme is discussed with some features in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

2.1 MANET Unicast Routing Protocols

Several routing protocols [12] have been proposed in recent years for possible deployment of Mobile Ad hoc Networks (MANETs) in military, government and commercial applications. In [13], these protocols are reviewed with a particular focus on security aspects. The protocols differ in terms of routing methodologies and the information used to make routing decisions. Four representative routing protocols are chosen for analysis and evaluation including: ad hoc on demand distance vector

routing (AODV), Dynamic Source Routing (DSR), Optimized Link State Routing (OLSR) and Temporally Ordered Routing Algorithm (TORA). Secure ad hoc networks have to meet five security requirements: confidentiality, integrity, authentication, non-repudiation and availability. Routing protocols for ad hoc wireless networks can be classified into three types based on the underlying routing information update as follows: **Reactive routing protocols** (on demand) obtain the necessary path, when required, by using a connection establishment process. Such protocols don't maintain the network topology information and they don't exchange routing information periodically. These protocols are such as DSR [14], The secure versions, such as, QoS Guided Route Discovery [15], Securing Quality of Service Route Discovery [16], Ariadne [17] and CONFIDANT [18], AODV [19], CORE [20], SAODV [21], SAR [22], TORA [23], SPREAD [24], and ARAN [25]. In **proactive or table driven routing protocols**, such as DSDV [26] or OLSR [27]. **Hybrid routing protocols** such as ZRP [28] and SRP [29] that combine the best features for both reactive and proactive routing protocols.

2.2 MANET Multicast Routing Protocols

There is a need for multicast traffic also in ad hoc networks. The value of multicast features with routing protocols is even more relevant in ad hoc networks, because of limited bandwidth in radio channels [30]. Some multicast protocols [31,32] are based to form and maintain a routing tree among group of nodes. Some other are based on to use routing meshes that have more connectivity than trees etc. It illustrates the main classification dimensions for multicast routing protocols as follows: **Multicast topology** [33] is classified into two approaches: mesh based and tree based [34,35]. Tree based approach is classified into two types; *Source tree based* and *Shared tree based*. Mesh based approach depends on multiple paths between any source and receivers pair. The mesh based protocols create the tree dependent on the mesh topology. **Routing initialization approach** is classified into three approaches namely source-initiated, receiver-initiated, and hybrid approach [36]. **Routing scheme** is classified into three approaches namely table-driven (proactive), on-demand (reactive), and hybrid approach [35,36]. **Maintenance approach** [36] is classified into two approaches namely softstate and hardstate.

2.3 Security Requirements

The security services of ad hoc networks are not different of those of other network communication paradigms. Specifically, an effective security paradigm must ensure the following security primitives: *identity verification*, *data confidentiality*, *data integrity*, *availability*, and *access control*. Although solutions to the above concerns have been developed and widely deployed in the wired domain, the amorphous, transient properties of ad hoc networks preclude their adaptation to server less network environments, which are often comprised of small devices. Instead, security solutions, in general, and key managements should strive for the following characteristics: ***Lightweight*:** Solutions must minimize the computation and communication processing to accommodate the limited energy and computational resources of ad hoc

enabled devices. **Decentralized**: Like ad hoc networks themselves, attempts to secure them must be ad hoc: they must establish security without a priori knowledge or reference to centralized, persistent entities. Instead, security paradigms must levy the cooperation of all trustworthy nodes in the network. **Reactive:** Ad hoc networks are dynamic: nodes trustworthy and malicious may enter and leave the network spontaneously and unannounced. Security paradigms must react to changes in network state; they must seek to detect compromises and vulnerabilities; they must be reactive, not protective. **Fault–Tolerant:** Wireless transfer mediums are known to be unreliable; nodes are likely to leave or be compromised without warning. The communication requirements of security solutions should be designed with such faults in mind; they mustn't rely on message delivery or ordering.

2.4 MANET Key Management Overview

MANET has some constrains such its energy constrained operations, limited physical security, variable capacity links and dynamic topology. So, there are different Key Management schemes are used to achieve the high security in using and managing keys. The crucial task in MANET uses different cryptographic keys for encryption like symmetric key, asymmetric key, group key and hybrid key (i.e. mixed of both symmetric key and asymmetric key). Here we discuss about some of the important Key Management schemes in MANET. **Symmetric Key Management:** the same keys are used by sender and receiver. This key is used for encryption the data as well as for decryption the data. If n nodes wants to communicate in MANET, k number of key pairs are required, where k=n(n-1)/2. Some of the symmetric key management schemes in MANET are Distributed Key–Pre Distribution Scheme (DKPS) [37], Peer Intermediaries for Key Establishment (PIKE) [38], and Key Infection (INF) [39]. **Asymmetric Key Management Scheme:** it uses two-part key. Each recipient has a private key that is kept secret and a public key that is published for everyone. The sender looks up or is sent the recipient's public key and uses it to encrypt the message. The recipient uses the private key to decrypt the message and never publishes or transmits the private key to anyone. Thus, the private key is never in transit and remains invulnerable. This system is sometimes referred to as using public keys. This reduces the risk of data loss and increases compliance management when the private keys are properly managed. Some of the asymmetric key management schemes in MANET are Self-Organized Key Management (SOKM) [40], Secure and Efficient Key Management (SEKM) [41], Private ID based Key Asymmetric Key Management Scheme [42]. **Group Key Management Scheme:** is a single key which is assigned only for one group of mobile nodes in MANET. For establishing a group key, group key is creating and distributing a secret for group members. There are specifically three categories of group key protocol (1) Centralized, in which the controlling and rekeying of group is being done by one entity. (2) Distributed, group members or a mobile node which comes in group are equally responsible for making the group key, distribute the group key and also for rekeying the group. (3) Decentralized, more than

one entity is responsible for making, distributing and rekeying the group key. Some important Group key Management schemes in MANET are Simple and Efficient Group Key Management (SEGK) [43], and Private Group Signature Key (PGSK) [44].
Hybrid Key Management Scheme: Hybrid or composite keys are those key which are made from the combination of two or more than two keys and it may be symmetric or a asymmetric or the combination of symmetric & asymmetric key. Some of the important Hybrid key management schemes in MANET are Cluster Based Composite Key Management [45], and Zone-Based Key Management Scheme [46].

2.5 Our Approach

In this paper, we propose the network model that contains some clusters; each cluster has its coordinator namely CH (initiator). The clusters are interconnected via CHs. There are subgroups of members called cluster in which one member is CH and virtual subgroup of CHs. Our model seems like CGSR [47] but in multicast manner. Our new key management scheme namely "Clustering based Group Key" (CGK) Management scheme that is a simple, efficient and scalable Group Key management for MANETs. Multiple tree based multicast routing scheme are used as mentioned in [48], which exploit path diversity for robustness. Also in [43], the author used two multicast trees for improving the efficiency and maintains it in parallel fashion to achieve the fault tolerances. So, in our scheme, two multicast trees are used for each subgroup (i.e. cluster subgroups or CHs' subgroup). For example, in a cluster, the connection of multicast tree is maintained be its CH that compute and distribute the intermediate keying materials to all members in this cluster through the active tree links. Also the CH is responsible for maintaining the connection of the multicast subgroup. In MANET, main cluster head namely MCH (its initiator) has the same CH role, but on the clusters' subgroup.

3 Our Group Key Management Scheme

3.1 Notations and Assumptions

Firstly, every node takes a valid certificate from offline configuration before entering the network. An underlying public key infrastructure is then used to manage certificates. However, many researchers are interesting of this hot topic, and most key management proposals suffer the man-in-the-middle attack. In this paper, each member has a unique identifier and all keying materials signed by CH in subgroup to make sure authenticity and integrity, in order to avoid the man-in-the-middle attack. Also, a group member has a password to join or can present a valid certificate. In our work, a group member can join by using a valid certificate. Here, for simplicity, we assume that a node can join a group if it has a valid certificate. Some notations used in CGK are listed in the table 1.

Table 1. Metric abbriviation

M_i	: i^{th} group member.	g	: Exponentiation base.
p	: Prime value.	CH_i	: i^{th} Cluster Head.
MCH	: Main Cluster Head.	N	: Total number of group members.
N_c	: Total number of Clusters.	n_{ci}	: Number of group members in i^{th} Cluster.
r_i	: A random number generated by i^{th} member, also called member private key.		
br_i	: Blinded i^{th} member key. ➔ $br_i = (g)^{ri} \bmod p$		
k_i	: Internal i^{th} member key, or intermediate key. ➔ $k_i = (b_{ri})^{ki} \bmod p$		
bk_i	: Blinded internal i^{th} member key, or blinded intermediate key. ➔ $bk_i = (g)^{ki}$		
K_{Gci}	: A key of i^{th} Cluster. ➔ $K_{Gci}=(br_{io})^{knci} \bmod p$		
K_G	: A key among CHs. ➔ $K_G=(br_{co})^{kNc} \bmod p$		

3.2 Overview of CGK

We proposed a new approach which aims to address the scalability problem while taking into consideration the dynamic aspect of the group members and dynamicity of nodes in MANET. There are two trees on the network to avoid the robustness problem as well. Our approach is based on clustering manner. Each cluster is initiated by CH, namely cluster initiator or coordinator initiator.

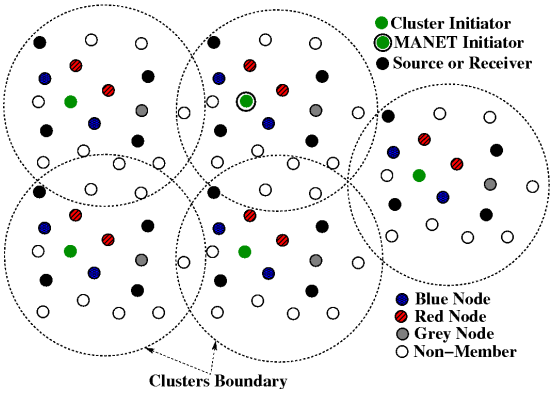


Fig. 1. MANET based on clustering

CH has then two keys; one for its cluster subgroup and another one for the interconnection among the clusters via CHs. Firstly, we describe our network model that is the mobile ad hoc network based on clustering that contains for example five clusters as shown in Fig. 1. There is a CH for each cluster and one of CHs is MCH. There are many multicast routing protocols have been proposed, these protocols are classified as shown before in section 2.2. We proposed another one in the category of multicast topology, tree-based and shared tree with double trees, namely Blue tree and Red tree. All clusters then works in parallel to construct two trees. Logically, a group member views the two trees as identical trees. The group members have to be in both multicast trees. **Inside the Cluster:** In a cluster, CH (initiator) starts to initialize the process for a cluster multicast subgroup by broadcasting a join advertises message across the

entire cluster. This cluster is bounded and having a fixed diameter. Each node is associated with three colors (blue, red, and grey). A node will choose its color (grey) when its total number of neighbors is less than a predefined threshold value (depending on average node degree, for instance, half of its degree). Other nodes randomly choose blue or red as their color with probability equal to 0.5. For the first received message, a grey node stores the upstream node ID and rebroadcasts the message except the node that the message is coming from. For a non-grey node, it stores the upstream node ID and rebroadcasts the message only if the upstream node is the same color, a sender/receiver, or a grey node. Based on the join response back from group members to CH, two multicast trees are formed in parallel, as shown in Fig. 2(a). It is noted that both trees consist of group members and intermediate non-member nodes. Sure both tree are constructed in parallel and in distributer processing manner, but in blue tree's point of view, we find that the red's nodes stop the broadcasting for blue tree and just blue's nodes who broadcasting the join advertises to both blue's nodes and grey nodes as shown in Fig. 2(b). As well, in red tree's point of view, we find that the blue's nodes stop the broadcasting for red tree and just red's nodes who broadcasting the join advertises to both red's nodes and grey nodes as shown in Fig. 2(c).

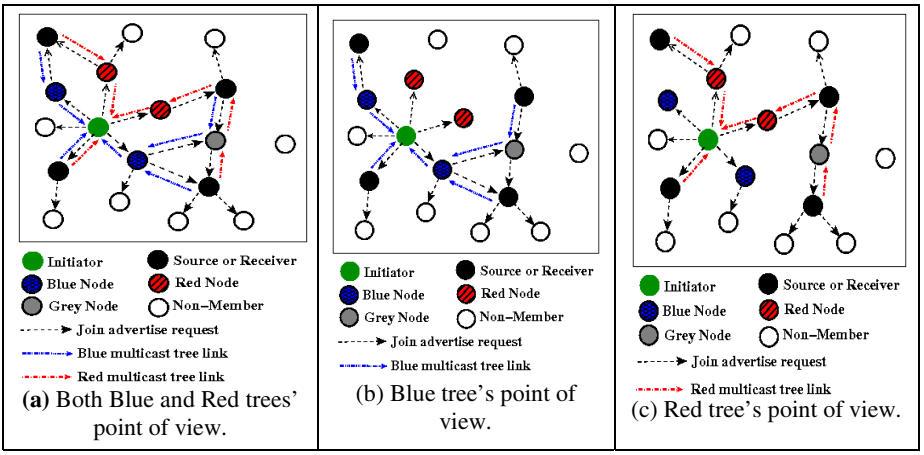


Fig. 2. Double multicast trees structure for a cluster

Interconnection among the Clusters: The interconnection among the clusters is via the MCH starts to initialize the process for a CHs' multicast subgroup by broadcasting a join advertises message across the entire MANET. We supposed the nodes no change its color, blue node still blue, red node still red, grey node still grey, and another CHs are source/receiver, viz the CHs seems as a virtual cluster. So we can apply the same scenario that is used before in the cluster, to get blue and red multicast trees among all CHs in MANET. This join advertises are broadcast across the entire network as shown in Fig. 8, in which the sequence number is used to avoid the loop, and the number of hops. Based on the join response back from CHs to MCH, two multicast trees are formed in parallel, as shown in Fig. 3. The double multicast trees among CHs are created and are shown in Fig. 4. Both trees consist of CHs, some of

group members, and intermediate non-group member nodes. The resultant two trees could be disjoint or may share a common node. As well, the double trees among CHs could be disjoint or may share some links in the double trees in the clusters. It is clear from the Fig. 5. Thus a dynamic double multicast trees structure for all is constructed as shown in Fig. 5. Initially MCH is responsible for sending the refreshment message periodically to maintain the connection of the double trees structure. After a predefined period of time, a member could decide to act a CH and notify the cluster members that it is on duty to maintain the cluster subgroup. As well, a CH could decide to act a main CH and notify the CHs that it is on duty to maintain the MANET group.

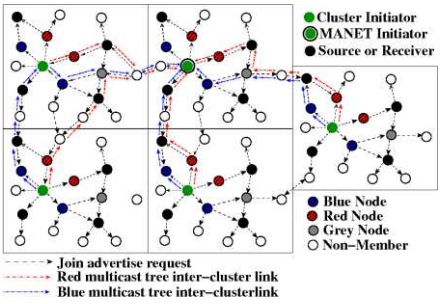


Fig. 3. Double multicast (Blue/Red) trees structure among CHs

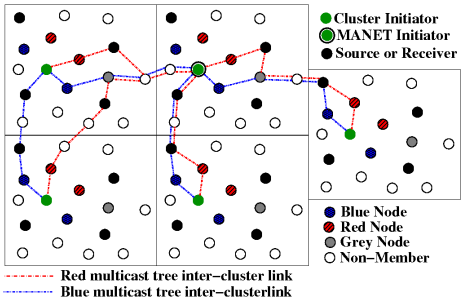


Fig. 4. CHs' multicast (Blue and Red) trees structure

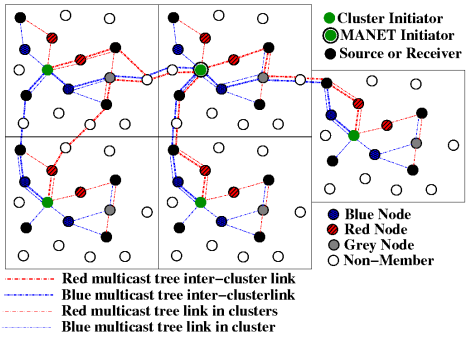


Fig. 5. Double multicast (Blue and Red) trees structure among all members in MANET

3.3 Multicast Group Management

A new member joins: A new member want to join a group, it could broadcast join requests to the group. The new member becomes a legitimate group member once its request is approved by any existing group member or by the CH of this group member. Any existing member can send replies back and send alarm "new member" to its CH. This CH then does the same procedure of handling join request that is similar to the above subgroup advertisement to ensure the consistency of the double multicast tree structure. **A member leaves:** The processing of handling members who leave is more complicated than handling the joining of new members. A leaving member will

not send a leaving notice. It leaves the group silently. Even if it could send a message and notify its leaving, this notice could get lost in a dynamic environment. There are a physical leaving and a logical leaving. For the physical leaving, a node moves out the range of the network or it switches its transmitter off. For a logical leaving, a node still stays inside the network, but it does not participate in the group activity. So there are two scenarios, as follows: *First scenario*: depends on detecting leaved members by its neighbors. Members are classified based in its places as follow: (1) Member is in the cluster double trees only, the neighbor of leaved member detect the leaved member and informs CH of its cluster to refresh the double multicast trees in this cluster. (2) Member is in CHs' double trees only, one of neighbor detects the leaving a member, then inform the MCH to refresh the double trees. (3) Member is in both a cluster double tree and CHs' double trees, a neighbor of leaved member detects that there is a member leaved, and inform both the MCH and its CH to refresh the double multicast trees of both CHs subgroup and the cluster of leaved member. *Second scenario*: is based on a "*member refresh*" message that is periodically broadcasted by CH across the subgroup. Each member should send an "*ack*" message back to indicate its status. The CH will determine whether a member remains attached or has left based on its response status within a certain time. If the cluster member on duty haven't receive "*member refresh*" message from its CA within a certain time, it sends a message "*I am CH*" and send refresh the double trees in the cluster, at the same time the MCH detects one CH leaved, so it refresh the double trees of CHs' subgroup and so on for the MCH, if it leaves. This scenario is quite more costly than the first scenario but is more appropriate for a highly dynamic network like MANET where the nodes move frequently and cause the connection to be broken frequently.

3.4 Group Key Establishment Protocol

The idea of subgroup key agreement protocol is that all subgroup members maintain a logic key's tree in local storage space. This key's tree is used to deduce the final common subgroup key.

Table 2. Members deduces locally the final common key

Inside M_1: $r_1 = 4$, $br_1 = g^{r1} \bmod p = 2^4 \bmod 13 = 3$, $k_1 = (br_1)^r \bmod p = 3^3 \bmod 13 = 1$, $bk_1 = g^{k1} = 2^1 = 2$ **$k_1 = (br)^{r1} \bmod p = (8)^4 \bmod 13 = 1$** $k_2 = (br_2)^{k1} \bmod p = (6)^1 \bmod 13 = 6$ $k_3 = (br_3)^{k2} \bmod p = (11)^6 \bmod 13 = 12$ $k_4 = (br_4)^{k3} \bmod p = (12)^{12} \bmod 13 = 1$ **$K_G = (br_0)^{k4} \bmod p = 6^1 \bmod 13 = 6$** **Inside M_2:** $r_2 = 5$, $br_2 = g^{r2} \bmod p = 2^5 \bmod 13 = 6$, $k_2 = (br_2)^{k1} \bmod p = 6^1 \bmod 13 = 6$, $bk_2 = g^{k2} = 2^6 = 64$ **$k_2 = (bk_1)^{r2} \bmod p = (2)^5 \bmod 13 = 6$** $k_3 = (br_3)^{k2} \bmod p = (11)^6 \bmod 13 = 12$ $k_4 = (br_4)^{k3} \bmod p = (12)^{12} \bmod 13 = 1$ **$K_G = (br_0)^{k4} \bmod p = 6^1 \bmod 13 = 6$**	**Inside M_3:** $r_3 = 7$, $br_3 = g^{r3} \bmod p = 2^7 \bmod 13 = 11$, $k_3 = (br_3)^{k2} \bmod p = 11^6 \bmod 13 = 12$, $bk_3 = g^{k3} = 2^{12} = 4096$ **$k_3 = (bk_2)^{r3} \bmod P = (64)^7 \bmod 13 = 12$** $k_4 = (br_4)^{k3} \bmod P = (12)^{12} \bmod 13 = 1$ **$K_G = (br_0)^{k4} \bmod p = 6^1 \bmod 13 = 6$** **Inside M_4:** $r_4 = 6$, $br_4 = g^{r4} \bmod p = 2^6 \bmod 13 = 12$, $k_4 = (br_4)^{k3} \bmod p = 12^{12} \bmod 13 = 1$, $bk_4 = g^{k4} = 2^1 = 2$ **$k_4 = (bk_3)^{r4} \bmod P = (4096)^6 \bmod 13 = 1$** **$K_G = (br_0)^{k4} \bmod p = 6^1 \bmod 13 = 6$**

Our scheme is based on key's tree structure, for each subgroup; there is individual key's tree and a common subgroup key. The key's tree structure (e.g. with 4 members included CH, as an example) in our scheme is shown in Fig. 6.Each member generates a private number; r1, r2, r3, and r4 for the members 1, 2, 3, and 4 respectively. CH of a cluster generates the numbers r and r0, and informs all other members in its cluster. The r, r0 at the two ends of the key tree for efficient group key refreshing and the CH role switching. Also, it is responsible for handling the member join and leave. All members reply its CH by intermediate keys to calculating keys. In this example: a subgroup contains four nodes. CH multicast the intermediated blind keys to all members. So, each member deduces locally the final common subgroup key. The given parameters' value for each node: g=2, p=13, r=3 then $b_r=g^r \bmod p=2^3 \bmod 13=8$, $r_0=5$ then $b_{r0}=g^{r0} \bmod p=25 \bmod 13=6$. Each M_i, $\forall$ iϵ[1,4], can calculate the K_G as shown in table 2.

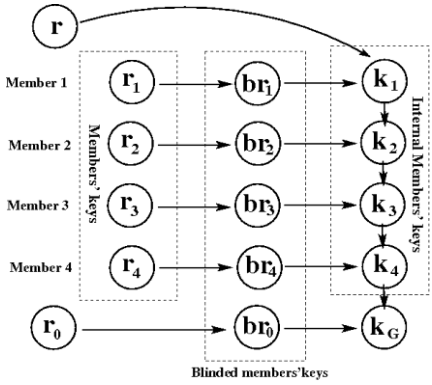


Fig. 6. Key's tree structure to generate group key (KG) with 4 members

Table 3. Deduceing the common key when member join

Inside M_1: $r_1 = 4$, $br_1 = g^{r1} \bmod p = 2^4 \bmod 13 = 3$, $k_1 = (br_1)^r \bmod p = 3^3 \bmod 13 = 1$, $bk_1 = g^{k1} = 2^1 = 2$ **$k_1 = (br)^{r1} \bmod p = (8)^4 \bmod 13 = 1$** $k_2 = (br_2)^{k1} \bmod p = (6)^1 \bmod 13 = 6$ $k_3 = (br_3)^{k2} \bmod p = (11)^6 \bmod 13 = 12$ $k_4 = (br_4)^{k3} \bmod p = (12)^{12} \bmod 13 = 1$ $k_5 = (br_5)^{k4} \bmod P = (3)^1 \bmod 13 = 3$ **$K_G = (br_0)^{k5} \bmod p = 6^3 \bmod 13 = 8$** **Inside M_2:** $r_2 = 5$, $br_2 = g^{r2} \bmod p = 2^5 \bmod 13 = 6$, $k_2 = (br_2)^{k1} \bmod p = 6^1 \bmod 13 = 6$, $bk_2 = g^{k2} = 2^6 = 64$ **$k_2 = (bk_1)^{r2} \bmod p = (2)^5 \bmod 13 = 6$** $k_3 = (br_3)^{k2} \bmod p = (11)^6 \bmod 13 = 12$ $k_4 = (br_4)^{k3} \bmod p = (12)^{12} \bmod 13 = 1$ $k_5 = (br_5)^{k4} \bmod P = (3)^1 \bmod 13 = 3$ **$K_G = (br_0)^{k5} \bmod p = 6^3 \bmod 13 = 8$**	**Inside M_3:** $r_3 = 7$, $br_3 = g^{r3} \bmod p = 2^7 \bmod 13 = 11$, $k_3 = (br_3)^{k2} \bmod p = 11^6 \bmod 13 = 12$, $bk_3 = g^{k3} = 2^{12} = 4096$ **$k_3 = (bk_2)^{r3} \bmod P = (64)^7 \bmod 13 = 12$** $k_4 = (br_4)^{k3} \bmod P = (12)^{12} \bmod 13 = 1$ $k_5 = (br_5)^{k4} \bmod P = (3)^1 \bmod 13 = 3$ **$K_G = (br_0)^{k5} \bmod p = 6^3 \bmod 13 = 8$** **Inside M_4:** $r_4 = 6$, $br_4 = g^{r4} \bmod p = 2^6 \bmod 13 = 12$, $k_4 = (br_4)^{k3} \bmod p = 12^{12} \bmod 13 = 1$, $bk_4 = g^{k4} = 2^1 = 2$ **$k_4 = (bk_3)^{r4} \bmod P = (4096)^6 \bmod 13 = 1$** $k_5 = (br_5)^{k4} \bmod P = (3)^1 \bmod 13 = 3$ **$K_G = (br_0)^{k5} \bmod p = 6^3 \bmod 13 = 8$** **Inside M_5:** (new Member) $r_5 = 4$, $br_5 = g^{r5} \bmod p = 2^4 \bmod 13 = 3$, $k_5 = (br_5)^{k4} \bmod p = 3^1 \bmod 13 = 3$, $bk_4 = g^{k4} = 2^1 = 2$ **$k_5 = (bk_4)^{r5} \bmod P = (2)^4 \bmod 13 = 3$** **$K_G$ =(br0)k5 mod p= 63 mod 13 = 8**

Initialization: CH announces its role and broadcasts two random keys (r, r_0) and its br_c, br, and br_0. Each member has unique identifier (ID) that is given by its CH when joining the group. At the initialization phase, the members are sorted by their ID. M_i, $\forall\ i\epsilon[1,N_c]$, (where N_c is number of subgroup's members) generates a private random number r_i then compute the br_i and send it to its CH. CH is then responsible for computing $k_1 \ldots k_{NC}$ and $bk_1 \ldots bk_{Nc}$ and then multicasts them to the subgroup's members. All keying materials are put in one package and the order of blinded intermediate key materials shows the structure of the key tree. Each member can thus deduce the subgroup key (K_G). **Member join:** new member can be easily added into the nearest cluster as described before in sec. 3.3. The double trees are constructed. CH insert the new member in the current rightmost position and give it ID. CH does not generate any random key but still provides key independence. Given blinded keys, new member deduces new common subgroup key, however it can't deduce the previous common subgroup key.

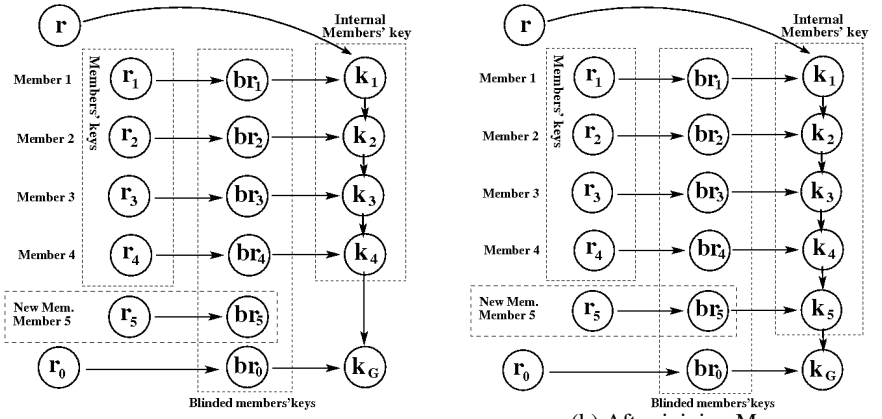


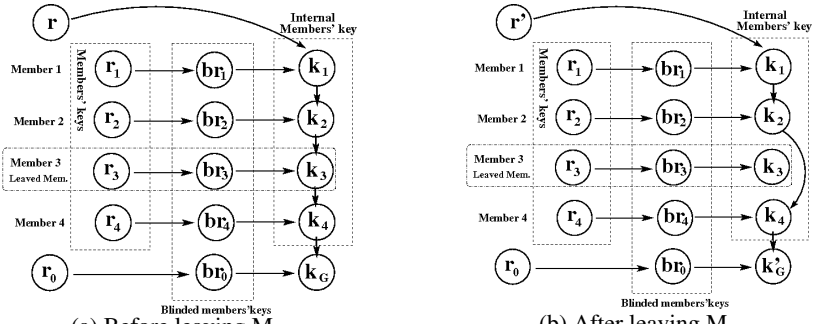
Fig. 7. Key tree structure to generate group key (KG), while a member join a subgroup

Table 4. Deduceing the common key when member leaves

Inside M_1: $r_1 = 4$, $br_1 = g^{r1} \bmod p = 2^4 \bmod 13 = 3$, $k_1 = (br_1)^r \bmod p = 3^5 \bmod 13 = 9$, $bk_1 = g^{k1} = 2^9 = 512$ **$k_1 = (br')^{r1} \bmod p = (6)^4 \bmod 13 = 9$** $k_2 = (br_2)^{k1} \bmod p = (6)^9 \bmod 13 = 5$ $k_4 = (br_4)^{k2} \bmod p = (12)^5 \bmod 13 = 12$ **$K_G = (br_0)^{k4} \bmod p = 6^{12} \bmod 13 = 1$**	**Inside M_2:** $r_2 = 5$, $br_2 = g^{r2} \bmod p = 2^5 \bmod 13 = 6$, $k_2 = (br_2)^{k1} \bmod p = 6^9 \bmod 13 = 5$, $bk_2 = g^{k2} = 2^5 = 32$ **$k_2 = (bk_1)^{r2} \bmod p = (512)^5 \bmod 13 = 5$** $k_4 = (br_4)^{k2} \bmod p = (12)^5 \bmod 13 = 12$ **$K_G = (br_0)^{k4} \bmod p = 6^{12} \bmod 13 = 1$** **Inside M_4:** $r_4 = 6$, $br_4 = g^{r4} \bmod p = 2^6 \bmod 13 = 12$, $k_4 = (br_4)^{k2} \bmod p = 12^5 \bmod 13 = 12$, $bk_4 = g^{k4} = 2^{12} = 4096$ **$k_4 = (bk_2)^{r4} \bmod P = (32)^6 \bmod 13 = 12$** **$K_G = (br_0)^{k4} \bmod p = 6^{12} \bmod 13 = 1$**

Fig. 7 depicts Key tree structure to generate group key (K_G), while new member wants to join a subgroup. We take the same previous example with adding new member M_5. The given parameters' value for each member: g=2, p=13, r=3 then $b_r=g^r$ mod $p=2^3$ mod 13=8, r_0=5 then $br_0=g^{r_0}$ mod $p=2^5$ mod 13=6. Each M_i, ∀ i ϵ[1, 5], can calculate K_G as shown in table 3. **Member leave:** Member can be easily leaved from its cluster as described before in sec. 3.3. The double trees are constructed. It is possible that the leaved member is either a member in a cluster or CH.

Case 1: leaving of a member in a cluster, its CH generates a new random key r' instead of r and multicast the blinded value br' as well as other intermediate blinded keys. Each M_i, ∀ i ϵ[1,Nc]\{leaved member}, can then calculate the K_{Gc}. *Case 2*: leaving of CH, a cluster member on duty acts as CH as before, moreover, the MCH detects a CH leaved, so the leaved process seems like two leaved members but really one leaved member, one from a cluster and another from the CHs'. In two cases, the leaved process simply takes place in a subgroup as shown in Fig. 8, that depicts key tree structure to generate both group key (K_{Gc}) for the cluster of leaved member and group key (K_G) for CHs via the same process, while a member leaves the multicast group. Also, we take the same example used before in this section with leaving M_3 in *Case 1*. The given parameters' value for each member: g=2, p=13, r'=5 then $br'=g^{r'}$ mod $p=2^5$ mod 13=6, r_0 =5 then $br_0=g^{r_0}$ mod $p=2^5$ mod 13=6. Each M_i, ∀ i ϵ[1,5]\{3}, can calculate the K_G as shown in table 4.**Group key refresh/reinforce: G**roup key may need to be changed periodically, and may not be related to any change of group membership. The purpose of refreshing the group key periodically is to prevent the long time use of group keys which could be compromised. This process can be implicitly done during the switch of CH, or explicitly performed by CH which generates a new random key r" and multicasts the blinded value br" as well as other intermediate blinded keys. Then each M_i,∀ i ϵ[1,N_c], can calculate the K_{Gc} as described in section 3.4. Refresh/reinforce process take place independently in each cluster, as well in the CHs' subgroup. That decreases the traffic overheads and increases the scalability in MANET.



(a) Before leaving M_3 (b) After leaving M_3

Fig. 8. Key tree structure to generate group key (KG), while member leaves

4 Discussion

The goal of all these protocols include such as minimal control overhead, minimal processing overhead, multi-hop routing capability, dynamic topology maintenance, loop prevention, or more secure. However many multicast routing protocols don't perform well in MANETs because in a highly dynamic environment, node move arbitrarily, and man-in-middle problem. Our paper focuses on the key management schemes that are important part of the security. So key management is an essential cryptographic primitive upon which other security primitives such as privacy, authenticity and integrity are built. As well, it has to be satisfied some features such as *Security*, *Reliability*, *Scalability*, and *Robustness*: **Security**: intrusion tolerance means system security should not succumb to a single, or a few, compromised nodes. So, key management schemes should ensure no unauthorized node receives key material that can later be used to prove status of a legitimate member of the network. Here a key is computed in distributed manner, and the member provides a trusted group communication. Other issues are trust management, vulnerability. Also, proper key lengths and cryptographic algorithms of adequate strength are assumed. **Reliability**: depends on key distribution, storage and maintenance and make sure that keys are properly distributed among nodes, safely stored where intruders aren't able to hack the keys and should be properly maintained. **Scalability**: key management operations should finish in a timely manner despite a varying number of nodes and node densities. It makes use the occupied network bandwidth of network management traffic as low as possible to increase nodes' density. **Robustness**: the key management system should survive despite Denial-of-Service attacks and unavailable nodes. Because of dynamicity of the group members, necessary key management operation should execute in a timely manner, in order not to make an isolated partition in the network. Multiple trees are used for robustness and avoid fault tolerance.

5 Conclusion

MANET is one of the most important and unique applications. Due to the nature of unreliable wireless medium data transfer is a major problem in MANET and it lacks security and reliability of data. A Key management is vital part of security. Key management protocols then play a key role in any secure group communication architecture. Moreover in MANET, members can join and leave the group dynamically during the whole session, plus the nodes movement. So, the key management is an important challenge because of its dynamism that affects considerably its performance. In this paper, we have studied the different key management schemes for MANET and proposed a new scheme namely CGK, which is an efficient/scalable hierarchical key management scheme for MANET multicast. In our scheme, the group members compute the group key in a distributed manner. This hierarchical contains two levels only, first level for all clusters' heads as a main group's members; the second level for all clusters' members. Then there is a secret key obtained in a distributed manner for each cluster subgroup, and another secret key for clusters' heads subgroup. It is

shown that our scheme reduces significantly the overall security overhead of member's join or leave compared to all other schemes and more reducing the ratio between control overheads and data.

References

1. Younis, M., Ozer, S.Z.: Wireless ad hoc networks: technologies and challenges. Wireless Communications and Mobile Computing 6(7), 889–892 (2006)
2. Guo, S., Yang, O.W.W.: Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. Computer Communications 30(9), 2129–2148 (2007)
3. Wang, J., Wang, C., Wu, Q. (eds.): Ad Hoc Mobile Wireless Network. National Defense Industry Press, Beijing (2004)
4. Xiao, C., Jie, W.: Multicasting techniques in mobile ad hoc networks. In: Mohammad, I., Richard, C.D. (eds.) Handbook of Ad Hoc Wireless Networks, pp. 25–40. CRC Press, Inc. (2003)
5. Junhai, L., Danxia, Y.: Research on routing security in MANET. Application Research of Computers 25(1), 243–245 (2008)
6. Renuka, A., Shet, K.C.: Hierarchical Approach for Key Management in Mobile Ad hoc Networks. Int. Journal of Computer Science and Information Security (IJCSIS) 5(1), 87–95 (2009)
7. Bouassida, M.-S., Chrisment, I., Festor, O.: Group Key Management in MANETs. International Journal of Network Security (IJNS) 6(1), 67–79 (2008)
8. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
9. Harn, L., Mehta, M., Wen-Jung, H.: Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA). IEEE Communications Letters 8(3), 198–200 (2004)
10. Phan, R.C.W.: Fixing the integrated Diffie-Hellman-DSA key exchange protocol. IEEE Communications Letters 9(6), 570–572 (2005)
11. Francis, M., Sangeetha, M., Sabari, A.: A survey of key Management Technique for Secure and Reliable Data Transmission in MANET. International Journal of Advanced Research in Computer Science and Software Engineering (IJAARCSSE) 3(1), 22–27 (2013)
12. Hongmei, D., Li, W., Agrawal, D.P.: Routing security in wireless ad hoc networks. IEEE Communications Magazine 40(10), 70–75 (2002)
13. Abusalah, L., Khokhar, A., Guizani, M.: A survey of secure mobile Ad Hoc routing protocols. IEEE Communications Surveys & Tutorials 10(4), 78–93 (2008)
14. Johnsort, D.B.: Routing in Ad Hoc Networks of Mobile Hosts. In: The First Workshop on Mobile Computing Systems and Applications, WMCSA 1994 (1994)
15. Wenjing, L., Wei, L., Yuguang, F.: SPREAD: Enhancing data confidentiality in mobile ad hoc networks. In: Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003. IEEE Societies (2004)
16. Hauser, R., et al.: Lowering Security Overhead in Link State Routing. Computer Networks 31(8), 885–894 (1999)
17. Yih-Chun, H., Perrig, A., Johnson, D.B.: Packet leashes: a defense against wormhole attacks in wireless networks. In: Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003. IEEE Societies (2003)

18. Sonja, B., Jean-Yves, B.: Performance analysis of the CONFIDANT protocol. In: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking Computing. ACM, Lausanne (2002)
19. Park, V.D., Corson, M.S.: A highly adaptive distributed routing algorithm for mobile wireless networks. In: Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1997. Driving the Information Revolution. IEEE (1997)
20. Sergio, M., et al.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. ACM, Boston (2000)
21. Garcia-Luna-Aceves, J.J., Spohn, M.: Source-tree routing in wireless networks. In: Proceedings of the Seventh International Conference on Network Protocols, ICNP 1999 (1999)
22. Yong, W., Attebury, G., Ramamurthy, B.: A survey of security issues in wireless sensor networks. IEEE Communications Surveys & Tutorials 8(2), 2–23 (2006)
23. Papadimitratos, P., Haas, Z.J.: Secure Routing: Secure Data Transmission in Mobile Ad Hoc Networks. In: ACM Workshop on Wireless Security, WiSe 2003, San Diego, California, USA (2003)
24. Lilien, L.: Developing Pervasive Trust Paradigm for Authentication and Authorization. In: Cracow Grid Workshop. Institute of Computer Science, AGH University of Science and Technology, Cracow,Poland Academic Computer Centre CYFRONET AGH (2004)
25. Jacquet, P., Muhlethaler, P., Qayyum, A.: Optimized Link State Routing Protocol. RFC 3626 (2003)
26. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, WMCSA 1999 (1999)
27. Clausen, T.H., et al.: The Optimized Link State Routing Protocol Evaluation through Experiments and Simulation. In: Proceedings of the IEEE Symposium on Wireless Personal Mobile Communications. Mindpass Center for Distributed Systems, Aalborg University, Fredrik Bajers Vej 7E, DK-9220 Aalborg, Denmark (2001)
28. Manel-Guerrero, Z.: Secure ad hoc on-demand distance vector routing. SIGMOBILE Mob. Computer Communications Review 6(3), 106–107 (2002)
29. Yih-Chun, H., David, B.J.: Securing quality-of-service route discovery in on-demand routing for ad hoc networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, Washington, DC (2004)
30. Chen, X., Wu, J.: Multicasting techniques in mobile ad-hoc networks. The Handbook of Ad-hoc Wireless Networks, 25–40 (2003)
31. Singh, T.P., Neha, Das, V.: Multicast Routing Protocols in MANETs. International Journal of Advanced Research in Computer Science and Software Engineering (IJAARCSSE) 2(1), 1–6 (2012)
32. Luo, J.: A survey of multicast routing protocols for mobile Ad-Hoc networks. IEEE Communications Surveys & Tutorials 11(1), 78–91 (2009)
33. Meghanathan, N.: Survey of Topology-based Multicast Routing Protocols for Mobile Ad hoc Networks. International Journal of Communication Networks and Information Security (IJCNIS) 3(2), 124–137 (2011)
34. Junhai, L., Liu, X., Danxia, Y.: Research on multicast routing protocols for mobile ad-hoc networks. Computer Networks 52(5), 988–997 (2008)
35. Siva, C., Murthy, R., Manoj, B.S.: Ad Hoc Wireless Networks Architectures and Protocols. Prentice Hall PTR, Upper Saddle River (2004)

36. Toh, C.K.: Ad Hoc Wireless Networks: Protocols and Systems, 1st edn. Prentice Hall PTR, Upper Saddle River (2001)
37. Chan, A.C.F.: Distributed symmetric key management for mobile ad hoc networks. In: The Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004 (2004)
38. Aziz, B., Nourdine, E., Mohamed, E.K.: A Recent Survey on Key Management Schemes in MANET. In: Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA 2008 (2008)
39. Anderson, R., Haowen, C., Perrig, A.: Key infection: smart trust for smart dust. In: Proceedings of the 12th IEEE International Conference on Network Protocols, ICNP 2004 (2004)
40. del Valle, G., Gómez Cárdenas, R.: Overview the key management in ad hoc networks. In: Ramos, F.F., Larios Rosillo, V., Unger, H. (eds.) ISSADS 2005. LNCS, vol. 3563, pp. 397–406. Springer, Heidelberg (2005)
41. Bing, W.: Secure and efficient key management in mobile ad hoc networks. Journal of Networks and Computer Applications 30(3), 937–954 (2007)
42. Anil, K., Sanjeev, R.: Identity-Based Key Management in MANETs using Public Key Cryptography. International Journal of Security (IJS) 3(1), 1–26 (2009)
43. Bing, W., Yuhong, J.W.: An efficient group key management scheme for mobile ad hoc networks. International Journal of Security and Networks (IJSN) 4(2), 125–134 (2009)
44. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
45. Pushpa, L., Kumar, A.V.A.: Cluster Based Composite Key Management in Mobile Ad Hoc Networks. International Journal of Computer Applications 4(7), 30–35 (2010)
46. Thair, K., Aref, A.: A Hybrid Schema Zone-Based Key Management for MANETs. Journal of Theoritical and Applied Information Technology (JATIT) 35(2), 175–183 (2012)
47. Chiang, C.-C.: Routing In Clustered Multihop, Mobile Wireless Networks With Fading Channel. In: Proceedings of IEEE SICON (1997)
48. Wei, W., Zakhor, A.: Multiple Tree Video Multicast over Wireless Ad Hoc Networks. IEEE Transactions on Circuits and Systems for Video Technology 17(1), 2–15 (2007)

Chord-Enabled Key Storage and Lookup Scheme for Mobile Agent-Based Hierarchical WSN

Alyaa Amer[1], Ayman Abdel-Hamid[2], and Mohamad Abou El-Nasr[1]

[1] College of Engineering and Technology
[2] College of Computing and Information Technology
Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt
{alya.amer,hamid,mnasr}@aast.edu

Abstract. It has been greatly acknowledged the emergence of the wireless sensor network (WSN) in many applications such as military, environmental and health applications. However, mobile agents have provided flexibility and customizability to overcome some of the WSN constraints such as limitation in power, computational capacities and memory through agent migration from node to node. Security is a crucial concern when it comes to mobile agents, due to threats from malicious hosts and other mobile agents, where the use of symmetric and asymmetric keys has been adopted to provide authentication and confidentiality. The use of asymmetric keys is nowadays feasible due to advances in WSN hardware. In this paper, Chord (A scalable peer to peer lookup service) is used for storing and looking up public keys in a clustered mobile agent WSN to protect sensor nodes from malicious agents. Cluster heads act as a distributed key storage and lookup facility forming a ring overlay network. Performance evaluation results through network simulation show that the proposed scheme provides efficiency and scalability in terms of key storage and lookup.

Keywords: Wireless Sensor Networks, Chord, Code Signing, Key Lookup, Mobile Agent, Distributed Hash Table.

1 Introduction

In past years, wireless sensor networks have drawn the attention of many researchers due to its high importance either in military or civilian applications such as environmental, traffic, industrial and agricultural monitoring. A wireless sensor network is an infrastructure less ad hoc network consisting of distributed small sized, low cost and power constrained sensor nodes named motes. The constraints associated with WSN lead to complexity in dealing with such network. A multitude of middleware approaches have been proposed to overcome some of these complexities by providing some features that can improve the performance of WSNs [1]. Examples of such middlewares are distributed database, message oriented, application driven, virtual machine and mobile agent (MA) middleware architectures [2]. Through the use of mobile agents, the sensor network

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 27–43, 2013.

can implement tasks as modules of the application helping consume less power, support multipurpose WSN and update network dynamics [1].

A Mobile agent is a computer program or software that migrate from a node to execute on another node on behalf of its dispatcher. Reliance on mobile agents is a promising approach that increases the utility of WSN due to the following: (1) the network overhead is decreased by moving the computation to the data not vice versa; (2) the network latency is lessened by using smart mobile agents that respond quickly to the changing environment in real time applications; (3) robustness and fault tolerance are increased by providing programmer control over self healing during node failure; (4) adaptation to user requirements, since new agents can be added with the required functionality [3]. An example of mobile agents' middleware is impala [4] and agilla, that allow the implementation of mobile agents in assembly language manner [5] [6].

Mobile agents have led to the existence of related security threats since the code moves from a node to another with its execution nature unlike the normal data transfer that does not affect the node by any means. Hence, security is an important concern when it comes to mobile agents migrating from a node to another in a WSN, where mobile agents may attack each other or the host itself and vice versa. However, authenticating these mobile agents requires requesting keys in every migration of the agent. Therefore, a mechanism is needed to get keys efficiently with a reduced overhead.

Since security keys are a main concern when it comes to security over WSN, an efficient and scalable key storage and lookup scheme is needed to accommodate the limited resources and constraints of sensor nodes. Therefore, this paper addresses such concern by employing an overlay network, where cluster heads (CHs) join a ring to store and lookup security keys using Chord algorithm [7]. Cluster heads joining this network are chosen as the nodes with the highest residual energy to maintain keys in a distributed hash table (DHT) with Chord-based load balancing. Hence, CHs act as a distributed key storage and look up facility. This paper proposes Chord-enabled key storage and lookup scheme for mobile agent-based hierarchical WSN scheme (CKSL-MA-HWSN) to assist in protecting a host from accommodating a malicious agent gaining unauthorized access to its resources and tampering with it, where in order to guarantee an agent's authenticity and protect its integrity, a digital signature technique is used to sign the agent's data.

The contributions of this paper can be summarized as follows. First, an efficient and scalable layer is added to lookup and store keys using Chord algorithm. Second, only cluster heads are in charge of implementing Chord to look up keys on behalf of their members to help conserve their resources, where all nodes public keys are stored at the cluster heads (according to Chord distribution). Third, keys are stored in DHT signed by the base station to assure their authenticity and integrity. Finally, cluster heads act as a distributed key storage and lookup facility for their cluster members to eliminate the overhead of communicating with a single centralized node or base station, acting as a single point of failure.

The rest of this paper is organized as follows. Section 2 highlights background and related work. Section 3 describes the proposed Chord-enabled key storage and lookup scheme. Section 4 illustrates performance evaluation results. Finally, the conclusion and the projected future work is covered in section 5.

2 Background and Related Work

This section outlines relevant background and related work. Section 2.1, describes mobile agents and their advantages and applications. Section 2.2, shows the mechanism and methodology of the Chord algorithm. Section 2.3, illustrates the related work in focus.

2.1 Mobile Agents and WSN

WSNs have lower bandwidth than wired networks, so the idea of employing mobile agents is beneficial, where the agent could perform all tasks locally on behalf of the user, eliminating redundancy to avoid data traffic exceeding the network capacity. Mobile agents have succeeded in intruding as an efficient technology in many applications proving their benefits, including e-commerce trading, distributed information retrieval, network awareness, network and systems management [8].

Mobile agents are computer programs or software that process data during their migration from node to another to perform some tasks on behalf of their dispatcher [9]. They are composed of three components: (1) code: program or software that is dispatched to perform a certain task on behalf of the dispatcher; (2) state: execution state of the running program; (3) data: data gathered as a result of the agent execution on the nodes. Agent migration is done through cloning or moving [10], it moves by carrying its state, data and code and resumes executing on the new node and no longer exist on the original node. Agent clones by copying its state, data and code to another node and resumes executing on both nodes. Mobile agents systems have added more capabilities to WSN by employing mobile agents that facilitates application re-tasking, local and information processing [9].

J. Baumann et al. [11] have shown that mobile agents have three modes of communication: (1) agent to node: agent accessing the data of the node it's moving or migrating to; (2) agent to agent: agents exchanging messages between them either locally or remotely; (3) node to agent: node accessing the resources of the agent residing on it. Despite the great additions of mobile agents to WSN technology, the presence of these agents dispatched by users with different objectives imposes some security threats. Mobile agents suffer from three types of security threats [12], agent to host threat where the agent gains unauthorized access to the host resources and tampers with it, agent to agent threat and host to agent security threat where the host compromises the agent residing on it, which is considered to be the most difficult attack to prevent since the host has a full control over the agent's code and its data. Recent work has been done

before in securing mobile agents and their host but not over WSN as seen in [13][14] i.e. code obfuscation, code encryption, white box cryptography, black box cryptography and code signing [13].

2.2 Chord Algorithm

Public keys based solutions provide more services and flexibility than symmetric keys [15], they provides simpler solution with much stronger security, therefore this paper focuses on protecting the node against malicious agents using asymmetric keys for signature generation and verification. Consequently, distributed hash table (DHT) is utilized to store all nodes public keys using Chord keys distribution, which provide scalable, efficient keys storage and lookup, where all lookups are resolved via *O(log N)* messages, N is the number of nodes in the network [7], thereby minimizing the memory overhead and decreasing the communication cost. There are many DHT based protocols e.g. Chord, CAN, Pastry, Tapestry, Freenet, Gnutella, Oceanstore, and Ohaha system, but Chord is distinguishable due to its simplicity, provable correctness and performance.

Nodes joining the Chord ring create a table named finger table with routing information about a small portion of the network nodes [7], each n node maintains information about $(n + 2^{i-1})$ nodes (all arithmetic is modulo 2^m), where $1 \leq i \leq m$, m is the maximum number of entries in the table, and the number of bits in the hashed IDs. Chord protocol is consistent since each node notifies its successor before leaving. Chord maintains its virtual topology by implementing three updates, the stabilize update, the notify update and the fix finger update, where stabilize and notify are used for newly joined and leaving nodes and fix finger is used to periodically update the finger table [16].

2.3 Related Work

In (C2WSN) [17] Chord protocol has been implemented for serving efficient queries in WSNs and in (CSN) [18] Chord has been utilized to efficiently locate the sensor nodes that stores a particular data item . Chord has also been used before for key establishment as seen in (CBKE) [19], to establish secret session keys between communicating nodes. But, symmetric keys provide less services than public keys. Although symmetric-key based schemes are widely used since they provide less computation complexity, but they have different weakness in scalability and connection probability. Moreover, (CBKE) does not make use of Chord algorithm to handle nodes leaving and joining the network. Related work shows that Chord protocol proved to be appropriate over WSN and its limited constraints and resources.

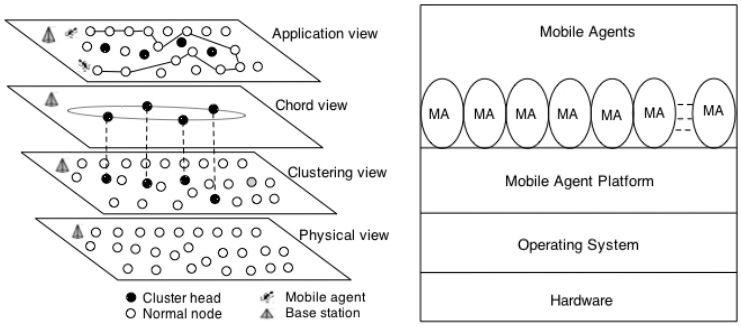
3 CKSL-MA-HWSN Scheme

The reliance on mobile agents in WSN and the associated security threats have motivated the idea of the proposed scheme, to provide a scalable, efficient and

flexible key storage and lookup scheme that helps nodes to check for agents authenticity and integrity as it moves from node to another. The idea of protecting the agent from the environment where it is executing is relatively a complex one, since the host has total control over the agent residing on it [14]. In this work, the problem of protecting the agent host against the malicious agent is addressed. Where the agent movement represents a threat on the host, since host and mobile code bear separate id entities. Therefore, the mobile agent's origin must be authenticated. Since, the mobile agent is exposed through the network. Thus, the host must verify the integrity of the agent it just received [13]. Consequently, this work aims at ensuring the authenticity and integrity of the mobile agents.

3.1 System View and Assumptions

CKSL-MA-HWSN scheme assumes a clustered WSN, where the Base station is aware of all nodes spatial location and their IDs.



(a) System View **(b)** Node View

Fig. 1. System and Node View

Figure 1a depicts the whole system view as a set of layers where the application layer contains a number of mobile agents dispatched by the BS to visit a number of nodes according to the task given. The application layer interacts with a logical layer (Chord layer) in terms of two messages to put a certain key *putkey* or get a key *getkey* and the Chord layer responds by *takekey* message. CHs only join this Chord layer and from Chord viewpoint, two nodes maybe viewed as neighbour nodes, while at the physical layer there are multiple hops in between them. Therefore, Chord does not consider the node's physical location. However, the CKSL-MA-HWSN scheme is built on a clustered wireless sensor network for the following reasons.

1. Clustered WSNs provide a better performance, in terms of routing, scalability and energy efficiency.

2. Only cluster heads are responsible for implementing the Chord keys lookup on behalf of other nodes (their cluster members) since they have a higher residual energy in order not to consume all nodes power and resources.
3. Cluster heads act as a distributed key storage and lookup facility, storing sensor nodes' keys to avoid having a single point of failure, target to attacks and decreasing the communication overhead of contacting a centralized node or the base station.

Finally, it has been proven that Chord's efficiency and scalability is remarkable when it comes to distributing large number of keys over smaller number of nodes [7].

The node model, shown in Figure 1b, is designed to give a detailed view of the node, where each node can support multiple agents. A mobile agent platform (MAP) is built over the host (OS and Hardware) and is responsible for the agents' arrival and departure [20] [21]. In addition, it performs two main tasks: security management task and key lookup task. The security management task is handled at normal nodes as a mobiles agent migrate from a node to another. The key lookup task is handled at cluster heads whenever a a certificate is needed. Where keys are preassigned to each MAP at each node with Chord distribution algorithm.

Before describing the operation of the CKSL-MA-HWSN scheme, the following assumptions are considered.

1. The WSN is clustered
2. Mobile agents are created, dispatched and signed only by the base station.
3. The base station sets the agent's itinerary before it is dispatched into the WSN.
4. The MAPs at all normal nodes and cluster heads are preloaded with their key pair (private and public keys) and base station public key at the initialization phase.
5. Chord algorithm does not include caching keys at sensor nodes.
6. The base station has a directory of all nodes' IDs and location.

3.2 Operation of the CKSL-MA-HWSN Scheme

Protecting the host by authenticating the mobile agent and protecting its code, data and state integrity is a main concern here. Such task is performed at each normal node by the mobile agent platform (MAP) using the method of code signing and verification as shown in Figure 2. The code signing technique uses digital signature and one-way hash function [14], therefore it does not reveal much about what the code can do and guarantees that the code is safe to use. During migration of mobile agents, they carry their code and data/state. The code carried is assumed to be static as received from the base station, but the data/state being transferred with the code from a node to another is dynamic as a result of the agents' task or execution outcome. Thus, the host protection process will be divided into two phases to be highlighted in the following subsections:

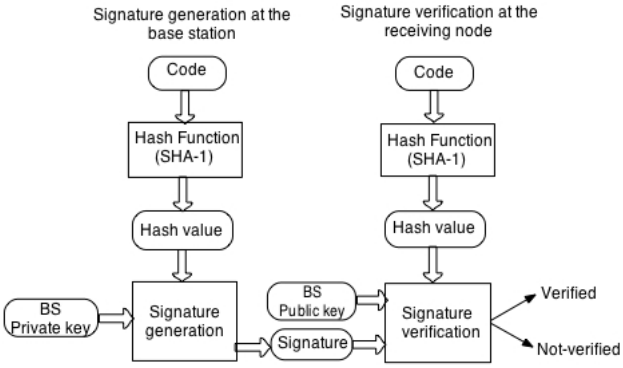


Fig. 2. Signing and verifying agent's code

phase 1 shows signing the code at the base station and verifying it at the receiving node, while phase 2 shows signing and verifying agent's data/state sent from node to node.

Phase 1: Signing the Verifying Agent's Code. In phase 1, the MAP at the base station signs the code after setting its itinerary and dispatches it to the target node. The agent starts visiting the target nodes to execute its code on each node on behalf of the dispatcher, returning the data collected back to the dispatcher, after executing on all nodes defined in its itinerary. Signature generation and verification is done using Elliptic curve cryptography (ECC) digital signature algorithm ECDSA, where ECC is used instead of RSA, since ECC offers the same security level as RSA but with smaller key size, less computations, memory overhead and bandwidth, which is more suitable for small devices [22,15]. Where 160-bit key is used for ECDSA, which is equivalent to 1024-bit key used for RSA level [23]. Secure hash algorithm (SHA-1) is used as the hash function for ECDSA generation and verification since it has better collision resistance than MD5 and MD4 but with a slightly higher cost of energy [22]. SHA-1 outputs a 160-bit digest of any sized input, it also has good distributional properties, which helps provide a load balance on the cluster heads, since nodes' IDs are hashed using SHA-1, IDs are generated with the same probability, therefore all cluster heads with high probability will receive the same number of keys according to Chord distribution. As the MAP at the destination node receives the signed hash as seen in equation (1) together with the agent's code, the MAP starts decrypting the signed hash and compares it with the hash produced after hashing the code it has received. If the two hashes are not the same, the code is proven to be unauthorized/illegal agent's code and will not have the authority to access the resources or execute on the target node. However, signing the code is done only once by the MAP of Base station (dispatcher) and verified at every receiving node's MAP.

$$S = E_{PR_{BS}}(H(C))||C \tag{1}$$

S: Transmitted Signature
$E_{PR_{BS}}$: Encryption with the base station private key
H: Hash function (SHA-1)
C: Agent's code

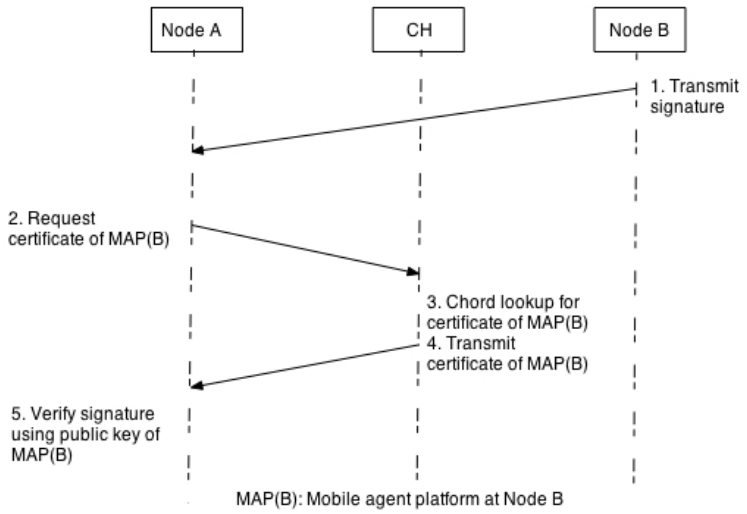


Fig. 3. Signing and verifying agent's data/state

Phase 2: Signing and Verifying Agent's Data/State Sent from Node B to Node A. In Phase 2 node to node authentication is required as seen in Figure 3 agent's data is changed from node to another according to the agent execution, in contrast to the static agent code that is only signed by BS. Phase 2 shows the utility and effectiveness of using Chord for the lookup of nodes' public keys. When the MAP at node A receives the signed data, it requests the certificate of MAP of node B from its cluster head which will in turn implement the Chord algorithm for key look up on behalf of A and returns the certificate of B in *O(log N)* as inspired by [7] (Where N here is the total number of nodes). MAP of node A decrypts the certificate received using the BSs public key to verify the authenticity and integrity of the received data. Cluster heads' role is remarkable, since they perform Chord key lookup operation on behalf of their cluster members. The CKSL-MA-HWSN scheme is robust in dealing with cluster heads failures, where keys are stored at each cluster head can be potentially repeated at the next two succeeding cluster heads to assure keys' availability in case of the nodes failure (this issue is left for future work). On the other hand, if a cluster head is voluntarily leaving the ring, it will transfer all the keys it is storing to its immediate successor. As a result, the new cluster head joining the ring searches for its successor, and through periodic functions, it will fix its finger table by asking its successor, maintain its keys and notify its successor of its presence.

3.3 Quantitative Analysis

This section focuses on quantifying the number of control messages in the system by focusing on three types of messages in the application level discarding the network level. *Findsucc* message: between CHs to get the successor of a certain key based on Chord protocol for key lookup. *Reqkey* message: between a node and its cluster head, to request the key of a certain node. *Getkey* message: between CHs to get a key from the hash table.

There are three different cases where these control messages are invoked as the agent moves from a node to another:

1. *Case 1*: This case is a 2 message cost (*Reqkey*) and its reply, where the receiving node discovers that its own cluster head stores in its DHT the public key of the sending node
2. *Case 2*: This case is a 4 messages cost, where the receiving node sends a *Reqkey* message to its cluster head and its reply, its cluster head discovers that its succeeding node holds the sender's public key or discovers directly from its finger table the node storing the sender's key, in both cases it will send a *Getkey* message to that node and also waits for the reply.
3. *Case 3*: This case is a 6 or more messages cost where receiving node sends a *Reqkey* message asking its cluster head for the sender's key and waits for the reply, the cluster head sends a *Findsucc* message to lookup that key according to Chord lookup, this *findsucc* message can be sent at least once until it reaches the cluster head holding the required key, at that point the receiving node's cluster head sends a *Getkey* message asking for the key and also waits for the reply.

According to the three cases, equation (2) shows a general case for calculating the total number of control messages and their replies in the system (M).

$$M = n_1 \times 2 + n_2 \times 4 + n_3 \times y \tag{2}$$

where n_1 is how many times the key was found at the node's the cluster head (*case 1*), n_2 is how many times the key was found at the cluster head's successor or the successor (holder) of the key was found directly in the cluster head's finger table (*Case 2*), n_3 is how many times (*Case 3*) was found, where the number of messages will be more than 2, y can be estimated as $3 \leq y \leq z$. Where $z = O(logN) + x$, z represents the maximum number of messages resulting from the agent's itinerary , $O(logN)$ maximum number of *Findsucc* messages that must be sent to find a successor in N node network according to Chord lookup protocol, x is the total number of times *Reqkey* message and *Getkey* message and their replies were invoked during the agent's itinerary. In general the exact value of y is dependent on the chord ring structure and the agent's itinerary.

3.4 Comparison to Related Work

Researches have been done to protect mobile agents and their host but not over WSN as seen in [14], [24], [25] but most of them are not suitable for WSN due to

its limited resources. Securing agent's host on WSN using public key cryptography was never addressed due to the assumption that public key cryptography is computationally infeasible over WSN. But some cryptographic algorithms like Elliptic curve cryptography (ECC) and Elliptic curve cryptography for digital signature (ECDSA) have proven to be efficient and viable over WSN as mentioned in [26,27]. The CKSL-MA-HWSN scheme offers such solution in addition to providing the Chord algorithm for storing and looking up public keys. Moreover, (CBKE) have also worked on using Chord algorithm for storing symmetric keys and establishing session keys. But, public keys provide more security services than symmetric keys, for example low storage, low communication cost and scalability. Therefore, the CKSL-MA-HWSN scheme has utilized Chord to efficiently store and lookup nodes' public keys in order to authenticate mobile agents' data. The CKSL-MA-HWSN scheme is also scalable since nodes can easily join and leave the network, avoiding the requirement that each node knows about every other node. In addition, it has low computation cost since ECC is used instead of RSA, which is more suitable for sensor nodes. Moreover, all lookups are resolved in *O(logN)*, so communication cost is decreased. Finally, this scheme offers high resilience since each node does not reveal much about other nodes in the network because every node stores a small portion of the keys.

4 Performance Evaluation

Performance evaluation results of the system have been conducted through experiments using the network simulator (NS-2) and AgentJ as a Java Virtual Machine (JVM) for the NS-2 simulation environment [28]. Section 4.1, shows the steps performed during the initialization phase. Section 4.2, validates the cases illustrated in (Section 3.3). Performance evaluation will be based on the following experiments: Section 4.3, shows the effect of MA itinerary on the number of control messages (in the application level). Section 4.4, illustrates the execution time of the cryptographic operations. Finally, section 4.5 shows the effect of MA itinerary on the average end-to-end delay.

Performance evaluation experiments will adopt the WSN shown in Figure 4 clustered according to the nodes' spatial location(4 clusters). Nodes 0, 8, 16, 24 are selected as the cluster heads, since they have the highest residual energy in their cluster. Each cluster head stores a number of keys according to its ID. For instance, cluster head 0 is storing in its DHT the certificate of nodes 25, 26, 27, 28, 29, 30 and 31 since these IDs lie between cluster head 0 and its predecessor (cluster head 24) in the Chord ring according to Chord protocol. Similarly, cluster head 8 will store the certificate of nodes 1, 2, 3, 4, 5, 6 and 7. Cluster head 16 will store the signed certificate of nodes 15, 14, 13, 12, 11 and 10. Cluster head 24 will store the certificate of nodes 17, 18, 19, 20, 21, 22 and 23 in its DHT.

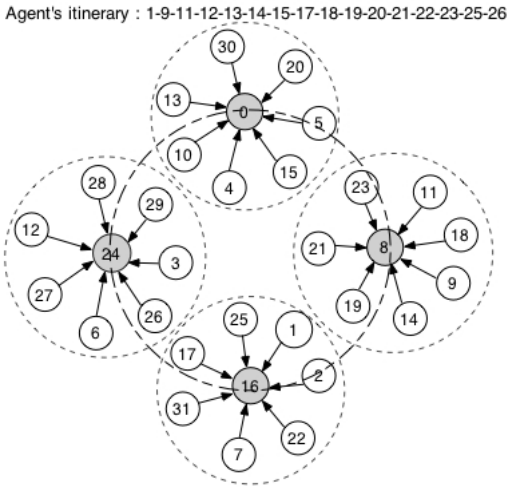


Fig. 4. Clustered WSN with CHs joining the Chord ring

4.1 Initialization Phase

The Chord ring is built up at the initialization phase. At each key storage in the DHT, the base station communicates with an arbitrary cluster head to lookup for the successor (holder) of that key it needs to store according to Chord pre-distribution. For example, in Figure 4 the base station asks cluster head 0 to store the certificate of node 22 in the DHT. According to the finger table of cluster head 0, it will ask cluster head 16 to lookup for 22, since it is the largest finger in its finger table that precedes 22, cluster head 16 will discover that its succeeding node 24 is the successor of the key of node ID 22, since 22 lies between 16 and 24, node 0 returns the ID of node 24 to the base station and the base station immediately puts the certificate of node 22 at node 24. Stabilize messages are frequently invoked to ensure that the finger tables and successor pointers are upto date. As a simple example, suppose node 4 wants to join the Chord ring, it will ask an arbitrary node for the successor of node 4, which is node 0, node 4 will be the successor of node 0 and will store the certificate of nodes 1, 2 ,3 and 4 in its DHT.

4.2 Results Validation

Based on Figure 4, the three cases mentioned in the quantitative analysis (section 3.3) can be deduced, for example, *Case 1* when an agent moves from node 1 to node 19 this costs the system only one control message i.e *Reqkey* and its reply, since the cluster head of node 19 stores the certificate of node 1. *Case 2* appears when an agent moves from node 1 to node 10 , this costs the system four control messages since node 10 will send *Reqkey* message to its cluster head 0 requesting the certificate of node 1 and waits for the reply, cluster head 0 will discover that its successor (cluster head 8) holds the certificate of node 1, it will send *Getkey*

message to cluster head 8 and waits for the reply, thus four control messages are invoked. *Case 3* here is emphasized by six control messages since the max of *Findsucc* message is 1 in addition to the reply . For instance, when an agent moves from node 1 to 26, this costs the system six control messages since node 1 will send *Reqkey* message to its cluster head 24 requesting the certificate of node 1 and waits for the reply, according to the finger table of cluster head 24, it will send to cluster head 0 a *Findsucc* message to query for the successor of the key and waits for the reply, then cluster head 0 will send a *Getkey* message to its successor (cluster head 8) and also waits for the reply, since node 1 lies between cluster head 0 and 8.

4.3 Effect of MA Itinerary on Number of Control Messages

Based on Figure 4, assume the agent's itinerary set by the base station includes the three mentioned cases (1→ 9→ 11→ 12→ 13→ 14→ 15→ 17→ 18→ 19→ 20→ 21→ 22→ 23→ 25→ 26), where the agent is to visit *N/2* nodes which is 16 node since *N= 32* (including node 1), where the agent's moves are independent of inter and intra-cluster. However, the agents itinerary shows inter and intra-cluster moves. Figure 5 shows that the number of control messages scales linearly as the number of visited nodes increases where at the end of the agent's itinerary after visiting 16 nodes, number of control messages has reached 36 control messages. (excluding the replies for each control message).

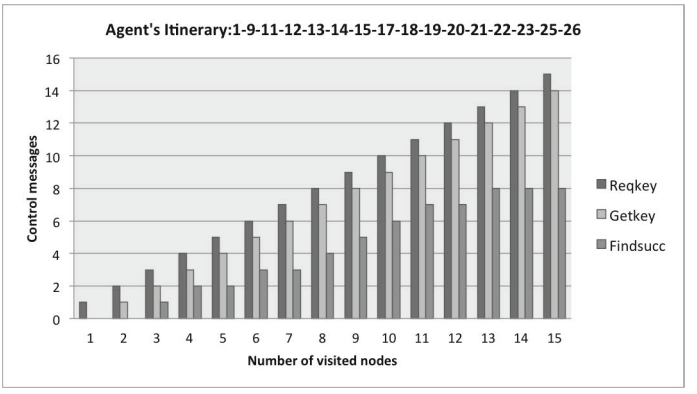


Fig. 5. Control messages Vs number of visited nodes

The results seen in Figure 5 can be validated using equation (2), where $n_1 = 2$, since *Case 1* appears 2 times from 1→ 9 and 15→ 17, where only *Reqkey* message is sent from the node to its cluster head. $n_2 = 5$, *Case 2* appears 5 times from 9→ 11, 13→ 14, 21→ 22, 23→ 25, 25→ 26, where *Reqkey, Getkey* messages are sent. $n_2 = 8$, since *Case 3* appears 8 times in the move from 11→ 12, 12→ 13, 14→ 15, 17→ 18, 18→ 19, 19→ 20, 20→ 21, 22→ 23 where *Reqkey, Findsucc* and

Getkey messages are sent, therefore $y = 3$ in this itinerary since the max number of messages shown in *Case 3* is 3 messages. Thus total number of messages as the agent reaches it's last destination is $M = (2 \times 1) + (5 \times 2) + (8 \times 3 = 36)$. Thus,36 control messages are invoked in the system excluding replies back, which is equivalent to the number shown in Figure 5 as the agent reaches its final destination (node 26). Thus, simulation results seen in Figure 5 are validated with the output obtained from equation (2).

4.4 Cryptographic Operations

The objective of this experiment is to show the overhead of cryptographic operations, Table 1 shows the execution time as data of 100 bytes is signed at node A (sender) and the certificate of node A together with the signature is verified at node B (receiver). Experiments were conducted for 10 times to get their average, on a processor Intel Core 2 Duo @2.4 GHz and system memory of 4GB. Key verification and signature verification are almost equivalent, since the key is 160 bit and the signed hash produced from SHA-1 is 160 bit. The resulting execution time shows an acceptable overhead concerning cryptographic operations.

Table 1. Execution time of cryptographic operation in sec

Operation	Average time
Signing data at node A	0.0434
Verifying the key (160 bit) of the sender at node B	0.0221
Verifying data received at node B	0.0224

In future, the power consumption of these operations will be investigated as inspired by [26][27][15].

4.5 Effect of MA Itinerary on the Average End-to-End Delay

The purpose of this experiment is to show the effect of the MA itinerary on the average end-to-end Delay. Assuming the agent has four different itineraries based on the WSN constructed in Figure 4 where:

1. *Itinerary 1*: Where all agent's itinerary steps result in only 1 control message (*Reqkey*) equivalent to *Case 1* mentioned in (section 3.3)
2. *Itinerary 2*: Where all agent's itinerary steps result in 2 control messages (*Reqkey*, *Getkey*) equivalent to *Case 2*.
3. *Itinerary 3*: Considered as the worst case, where all agent's itinerary steps result in 3 control messages.
4. *Itinerary 4*: Consists of the 3 different cases where some itinerary steps result in 2, some result in 3 and others result in only 1 control message *Reqkey*, *Getkey*, *Findsucc*)) (see Figure 5 for Agent's itinerary).

In all the four itineraries the agent visits $\lceil N/2 \rceil$ nodes where *N=32* (including the first node receiving the agent from the BS) where agent's itinerary is sent by the base station. End-to-end delay for the three different itineraries was repeated for 10 times to get the average end-to-end delay, where it was calculated starting from the time the source node signs the agent upto the last node (destination) verifies the agent and replies with a message . This average end-to-end delay is verified to be within 95% confidence interval for the four itineraries in three different scenarios.

1. *Scenario 1*: Sending an agent from node to another without cryptographic operations done at sender or receiver (base scenario).
2. *Scenario 2*: Sending a signed agent, where the receiving node consumes time verifying the agent, assuming sender's public key is already maintained at the receiving node. (cryptographic operations include code, data/state signing and verification overhead, however no key verification overhead included)
3. *Scenario 3*: Sending a signed agent, where the receiving node will search for the sender public key and then verifies the agent.(cryptographic operations as in *scenario2* in addition to key lookup and verification overhead).

Table 2. Main operations in steps for Scenario2 and Scenario3

Scenario 2	Scenario 3
Operations and estimate time/itinerary step: A signs agent (x_1) A sends agent and received by B (Nt_1) B verifies signature(x_2) B replies (Nt_2)	Operations and estimate time/itinerary step: A signs agent(x_1) A sends agent and received by B (Nt_1) B lookup key of A (L) B verifies Key of A(x_2) B verifies signature(x_3) B replies (Nt_2)
Estimate time/itinerary: (N-1)* ($\sum x_i$+Nt_1 +Nt_2)	Estimate time/itinerary: (N-1)* ($\sum x_i$+Nt_1 +Nt_2+L)

Table 2 shows analysis of the main operations involved in one step of the itinerary for *Scenario2* and *Scenario3*, where x_i: is time measurement, and assuming an agent is sent from node A (source) to node B (destination) representing one step in the itinerary, where the agent's itinerary includes *N-1* nodes. In order to verify the results obtained from this experiment. For each operation an estimate run time is assumed. Finally, the estimate run time per itinerary is also calculated.

Table 3 shows percentage ratios, where ratios obtained in Scenario2 are in references to values obtained in Scenario1. For instance, see equation (3) where Ratio of (*Scenario i/Itinerary j*)=(R_{S_i,I_j}) and Value of (*Scenario i/Itinerary j*)=(V_{S_i,I_j}).

$$R_{S_i,I_j} = (V_{S_i,I_j} - V_{S_{i-1},I_j})/V_{S_i,I_j} * 100 \quad (3)$$

For example $R_{S_2,I_1} = (0.8327 - 0.3898)/(0.8327) = 53.2\%$. Similarly, $R_{S_3,I_1} = (0.8429 - 0.8327)/(0.8429) = 1.2\%$. Values shown in the Table 3 represent a

Table 3. Average end-to-end delay in sec and percentage ratio

Scenario \ Itinerary	1		2		3		4	
	Value	Ratio	Value	Ratio	Value	Ratio	Value	Ratio
1	0.3898	-	0.023661	-	0.026354	-	0.3521	-
2	0.8327	53.2%	0.9122	61.4%	0.7957	46.4%	0.7618	53.7%
3	0.8429	1.2%	1.1125	18%	1.3502	41.1%	1.0393	26%

real implementation in Java, integrated with NS-2 using AgentJ, to calculate the average end-to-end delay as the agent moves from node to another in the 3 scenarios for the 4 itineraries,, where this time was calculated at the moment the MAP at source node signs an agent until the MAP at destination node verifies this agent and replies back with *received* message in case *Scenario 1* or replies with *verified* or *not verified* message in case of *Scenario 2* or *Scenario 3*. Table 3 ratios shown in *Scenario2* represent the overhead of adding cryptographic operations where this overhead ranges from 49% to 61%. In *scenario3*, R_{S_3,I_1} represents the minimum overhead, since the key lookup process resulted in *1 control message*, R_{S_3,I_3} represents the maximum overhead in case of adding the key lookup process since I_3 is the itinerary where each itinerary step results in *3 control messages*. Finally, R_{S_3,I_4} is an intermediate percentage between the four itineraries since it includes the *3 cases* mentioned in section 3.3. In conclusion, the overhead of adding Chord lookup in the 4 itineraries is low in comparison to the overhead obtained from cryptographic operations, which is still acceptable overhead in order to obtain the security needed. In future, we plan to compare key lookup using Chord with lookup keys from centralised node (BS).

5 Conclusion and Future Work

This paper has provided two main security requirements, agent authentication and integrity to protect the host against malicious agents accessing its resources and tampering with it. However, protecting the agent's data has motivated the use of Chord algorithm for storing and looking up keys in an efficient and scalable way to avoid consuming the node resources in terms of memory and power, where a small portion of keys are stored at each cluster head offering high resilience to node capture, signing those keys stored at the cluster head with the base station public key has ensured their integrity. Also, Keys availability is also attained since each departing node notifies its successor of its departure in case of shortage in resources and all keys are transferred to its successor. Finally, the CKSL-MA-HWSN scheme is scalable, since communication cost scales logarithmically with the number of Chord nodes.

This work can be continued by adding key revocation and replication and another security services such as providing agent confidentiality during its movement or cloning from node to another. Future work also includes highlighting in details the operation of the proposed scheme in case of cluster heads joining, leaving and failure.

References

1. Henricksen, K., Robinson, R.: A survey of middleware for sensor networks: State-of-the-art and future directions. In: Proceedings of the International Workshop on Middleware for Sensor Networks, pp. 60–65 (2006)
2. Tong, S.: An evaluation framework for middleware approaches on wireless sensor networks.Technical report, Helsinki University of Technology (2009), http://cse.tkk.fi/en/publications/B/5/papers/tong_final.pdf
3. Chen, M., Kwon, T., Yuan, Y., Leung, V.C.M.: Mobile Agent Based Wireless Sensor Networks. Journal of Computers 1(1), 14–21 (2006)
4. Qi, H., Iyengar, S.S., Chakrabarty, K.: Multiresolution data integration using mobile agents in distributed sensor networks. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31(3), 383–391 (2001)
5. Daniel, T.: Cogent computing: information extraction from large-scale wsns- a complex querying perspective. Technical report number COGENT.002, Coventry university (2008)
6. Fok, C.-L., Roman, G., Lu, C.: Mobile agent middleware for sensor networks: an application case study. In: Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, pp. 382–387 (2005)
7. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R.: Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking, 17–32 (2003)
8. Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C., Pantziou, G.: Mobile Agent Middleware for Autonomic Data Fusion in Wireless Sensor Networks. In: Autonomic Computing and Networking, pp. 57–81 (2009)
9. Chen, M., Gonzalez, S., Leung, V.C.M.: Applications and design issues for mobile agents in wireless sensor networks. IEEE Trans. on Wireless Communications 14, 20–26 (2007)
10. Fok, C.-L.: A mobile agent middleware for wireless sensor networks (2008), http://mobilab.wustl.edu/projects/agilla/
11. Baumann, J., Hohl, F., Radouniklis, N., Rothermel, K., Straβer, M.: Communication Concepts for Mobile Agent Systems. In: Rothermel, K., Popescu-Zeletin, R. (eds.) MA 1997. LNCS, vol. 1219, pp. 123–135. Springer, Heidelberg (1997)
12. Vijil, E.C.: Security issues in mobile agents. Master's thesis, Indian Institute of Technology, Bombay (2002), http://etd.aau.edu.et/dspace/bitstream/123456789/1552/1/Tinbit%20Admassu.pdf
13. Sharma, S., Patheja, P.S., Waoo, A.A., Gour, R.: A survey on different security techniques of mobile code. International Journal of Engineering and Advanced Technology (IJEAT) 1(1) (2011)
14. Borselius, N.: Mobile agent security. IEEE Electronics and Communication Engineering Journal 14(5), 211–218 (2002)
15. Wander, A.S., Gura, N., Eberle, H., Gupta, V.: Energy analysis of public-key cryptography for wireless sensor networks. In: Third IEEE International Conference on Pervasive Computing and Communications, PerCom, pp. 324–328 (2005)
16. Baqer, M., Khan, A.I., Baig, Z.A.: Implementing a graph neuron array for pattern recognition within unstructured wireless sensor networks. In: Enokido, T., Yan, L., Xiao, B., Kim, D.Y., Dai, Y.-S., Yang, L.T. (eds.) EUC Workshops 2005. LNCS, vol. 3823, pp. 208–217. Springer, Heidelberg (2005)
17. Yu, J., Liu, W., Song, J.: C2WSN: A two-tier chord overlay serving for efficient queries in large-scale wireless sensor networks. In: International Conference on Advanced Computing and Communications, ADCOM 2007, pp. 237–242 (2007)

18. Ali, M., Uzmi, Z.A.: Csn: a network protocol for serving dynamic queries in large-scale wireless sensor networks. In: Second Annual Conference on Communication Networks and Services Research, May 19-21, pp. 165–174 (2004)
19. Zhang, F., Shi, Z.J., Wang, B.: Chord-based key establishment schemes for sensor networks. In: Fifth International Conference on Information Technology: New Generations, ITNG 2008, pp. 731–737 (2008)
20. Usman, M., Muthukkumarasamy, V., Wu, X.-W., Khanum, S.: Securing mobile agent based wireless sensor network applications on middleware. In: International Conference on Communications and Information Technologies (ISCIT), October 2-5, pp. 707–712 (2012)
21. Fok, C.-L., Roman, G.-C., Lu, C.: Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. ACM Trans. Auton. Adapt. Syst. 4, 16:1–16:16 (2009)
22. Potlapally, N.R., Ravi, S., Raghunathan, A., Jha, N.K.: Analyzing the energy consumption of security protocols. In: Proceedings of the 2003 International Symposium on Low Power Electronics and Design, ISLPED 2003, August 25-27, pp. 30–35 (2003)
23. Jansma, N., Arrendondo, B.: Performance Comparison of Elliptic Curve and RSA Digital Signatures; Technical Report; University of Michigan, Ann Arbor, MI, USA (2004)
24. Ametller, J., Robles, S., Ortega-Ruiz, J.A.: Self-protected mobile agents. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. (1), pp. 362–367 (2004)
25. Ismail, L.: A secure mobile agents platform. Journal of Communications 3(2) (2008)
26. Piotrowski, K., Langendoerfer, P., Peter, S.: How public key cryptography influences wireless sensor node lifetime. In: Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 169–176 (2006)
27. Meulenaer, G., Gosset, F., Standaert, O.-X., Pereira, O.: On the energy cost of communication and cryptography in wireless sensor networks. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WIMOB 2008, pp. 580–585 (2008)
28. Taylor, I.: AgentJ Java Network Simulations in NS-2. A Protean Research Group Project Naval Research Laboratory (2010), `http://downloads.pf.itd.nrl.navy.mil/archive/agentj/agentj.pdf`

Hardware Advancements Effects on MANET Development, Application and Research

Amr ElBanna[1], Ehab ElShafei[1], Khaled ElSabrouty[1], and Marianne A. Azer[1,2]

[1] Nile University, Cairo, Egypt
{amr.banna,ehab.elshafei}@nileu.edu.eg, sabrouty@gmail.com
[2] National Telecommunication Institute, Cairo, Egypt
mazer@nileuniversity.edu.eg

Abstract. Mobile devices' development has remarkably improved in light of the fast growing hardware advancements. These advancements include multi-core processor chips, ultra large main memories and batteries that last for hours even when running modern applications such as file transfer, voice communication and video streaming … etc. In this paper, we shed the light on recent and future trends of hardware advancements for mobile devices, and their impact on MANET developments. In addition, the effect of such advancements is investigated on application and different research areas.

Keywords: Ad hoc Networks, batteries, processing power, mobile devices capabilities.

1 Introduction

Mobile Ad hoc NETworks (MANETs) have been considered a worldwide trend in the past few decades. MANETs do not depend on centralized infrastructure; their strength is in using mobile wireless devices. MANET devices communicate directly with each other when they are within the same communication range. Otherwise, they rely on their neighbors to route messages. Due to the open medium and wide distribution of devices, MANETs are vulnerable to a wide range of security threats. In the early days, MANETs' wireless devices, such as laptops, PDAs or mobile phones, have limited processing capabilities and power resources. Therefore, developing lightweight protocols and security mechanisms were considered as a challenge.

The extensive use of mobile devices has phenomenally pushed the limits of hardware development in micro-processing devices. Exploring the usage of Graphics Processor Unit (GPU) as a general-purpose co-processor to accelerate compute-intensive applications has been an active research subject in the past few years [1]. This can be noticeably seen at non-professional end users in playing games, capturing and editing videos scenes, or even more in watching HD or 3D videos. On top of that, large enterprises are working very hard to provide ubiquity solutions to their industry professionals to be at their fingertips. This is to cope with the rapidly growing market

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 44–53, 2013.

and to flatten all hurdles that could delay business from going forward. We can obviously see this now in common solutions that were implemented to let professionals interact with their emails, chat, or do minimal jobs with their business colleagues wherever they are. This helps mobile market to continuously expand. This has also encouraged mobile manufacturers to build mobile devices as general business computers; in order to replace the desktop or even small or medium scale servers. The experimental investigations in [1] confirm that a mobile GPU, although designed primarily for low power rather than maximum performance, can provide significant performance speedup for vision tasks on a mobile platform. This is similar to the role of its high performance counterparts in the desktop and server systems. In this paper, we focus on recent hardware advancements for mobile devices and their impact on applications and different research areas of MANETs.

The remainder of this paper is organized as follows. Section 2 focuses on recent Advancement in hardware. Practical implementations of such hardware advancements and their impacts on MANETs research are presented in section 3 and section 4 respectively. Finally, in section 5, we conclude this paper.

2 Recent Hardware Advancements

Throughout the last few years, mobile devices' hardware has been subject to noticeable improvements. In section A and B, the processing and batteries advancements are presented respectively.

2.1 Processing Advancements

In this section, we conduct a comparison between most recent mobile phone devices versus some old ones developed five years ago by the same manufacturers. The results of our comparison from [2] are summarized in Table 1. We selected two of the biggest manufacturers in mobile devices (Apple & Samsung) and picked two mobile phones for each manufacturer. One released in year 2007 and the most recent one that released in 2012.

From Table 1, we can see that the processing capabilities in Apple increased by around 315% more than its five years counterpart. Similarly, this comparison shows the processing capabilities developed even more and increased by 424%. This is apart from other noticeable advancements in other components such as (screen, memory, GPS, batteries … etc.).

Nonetheless, [3] shows the giant mobile manufacturer "Samsung" licensed two 64bit processors designs; it signed in a contract with ARM, a British company that is considered as one of the biggest companies for developing processors. The magazine also mentions that faster 64-bit processor will appear in servers, high-end smartphones and tablets. Hence, we can anticipate remarkable turn over in micro-processing advancements.

Table 1. Comparison between mobiles manufactured in 2007 & 2012 by Apple and Samsung

Company	**Apple**		**Samsung**	
Year	**2007**	**2012**	**2007**	**2012**
Model	iPhone	iPhone 5	i450	Galaxy S3
CPU	412 MHz ARM 11	1.3 GHz Apple A6 (Dual Core Apple Swift)	330 MHz ARM 1136	1.4 GHz Cortex-A9 by ARM (Quad-core)
GPU	PowerVR MBX	PowerVR SGX 543MP3 (triple-core graphics)	PowerVR MBX	Mali-400MP
Internal Mem.	4/8/16 GB	16/32/64 GB	40 MB	16GB, 32GB, 64GB
RAM	128 MB[1]	1 GB	Null	2 GB
WLAN	WiFi 802.11b/g	WiFi 802.11 a/b/g/n, dual-band, WiFi hotspot	Null	WiFi 802.11 a/b/g/n, DLNA, WiFi Direct, WiFi hotspot
Battery	Standard battery, Li-Ion	Standard battery, Li-Po 1440 mAh (5.45 Wh)	Standard battery, Li-Ion	Standard battery, Li-Ion 2100 mAh
GPS	Null	Yes	Null	Yes

2.2 Battery Advancements

Mobile devices have witnessed a huge leap and technological advancements over the years, mainly thanks to the advancement in processors and memory modules. However, there's always been a sort of a bottleneck in mobile devices development; and that is power consumption and battery capacity.

Development of batteries' capacity is not following the same pace as the processors (according to Moore's law) [4]. Figure 1 [2] shows a comparison between three market leaders in manufacturing mobile phones. It shows batteries advancements have growing exponentially for last few years. However, in light of the recent developments, we believe that batteries capabilities are adequate to implement some practical applications based on 802.11 Ad Hoc networking.

The Smartphones now have evolved to encompass different type of applications and circuitry. A combination of these applications can function simultaneously very well for at least good four hours on Samsung Galaxy, Nexus, HTC, or iPhone for example. If we consider a real life scenario of using ad hoc networking for collaboration between users in a class session or a business meeting, these four hours can be good enough. We can also consider a scenario of inter vehicles communication on

[1] Apple provides no information regarding the RAM used in the iPhone, but software analysis has confirmed that it has 128 MB onboard

roads as to enhance the security and traffic jams prediction, where the drivers' smartphones would gather, analyze and share data. Four or five hour per day is again quite adequate considering that the average driving hours per day are four hours in a city like Cairo. Another scenario would be mobile games, and games tournaments; where multiplayer games would be installed on the players' mobile devices and teams can form on the spot and start to compete.

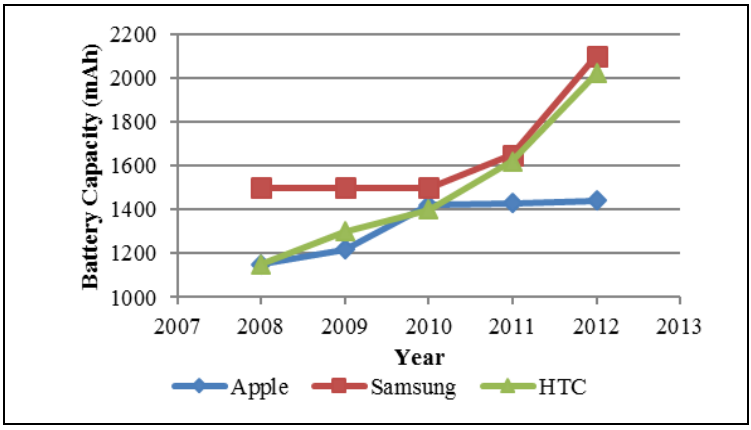


Fig. 1. Illustrated the battery manufacturing advancement made over year in terms of capacity per different vendors [2]

Most of the Smartphones nowadays support many types of wireless technologies, especially: 3G, WiFi and Bluetooth. We are focusing mainly on the power utilization over WiFi (802.11). The standard has different versions, each with a different power utilization profile. Usually, battery capacity is measured in milliAmpere/Hour. It is important to assess for how long, in terms of hours, would the devices be able to remain functional? Table 2 [5] illustrates the different versions of the 802.11 protocol and the associated speed and power consumption.

Table 2. 801.11 Versions and Power Consumption [5]

Standards	Range (m)	Speed (Mbps)	Power Consumption
802.11a	120	54	TX 510 mA@ 3.3 V
802.11b	140	11	TX 380 mA@ 3.3 V
802.11g	140	54	TX 400 mA@ 3.3 V
802.11n	250	600	TX 450 mA@ 3.3 V

The Nokia Energy Profiler is an application running on the mobile device that allows making measurements without any external hardware. It provides the values for power, current, temperature, signal strength and CPU usage. In Table 3 [6], researchers used a Nokia N95 smartphone to measure the power utilization for different wireless communications technologies. The team in [6] has compared results obtained

with the energy profiler with the ones obtained with the Agilent 66319D and found no significant difference between those two. Agilent is a hardware device that offers several features ideal for testing wireless and battery powered devices. A node in ad hoc networks can bear the responsibility of sharing internet connections or acting as a gateway to another type of networks. The same paper covered these results. Figure 2 [6] illustrates the power utilization as measured from a mobile phone utilizing an individual communication technology or the equivalent summed up energy consumed for utilizing a combination of these technologies, e.g. WiFi, UMTS, Bluetooth, … etc.

Table 3. Measures for Wireless Power Consumption [6]

Technology		Action	Power [mW]	Energy [J]
Wireless Data	Bluetooth	BT off	12	
		BT on	15	
		BT connected and idle	67	
		BT discovery	223	
		BT receiving	425	
		BT sending	432	
	WiFi IEEE802.11 (infrastructure mode)	In connection	868	8.2
		In disconnection	135	0.4
		Idle	58	
		Idle in power save mode	26	
		Downloading@4.5Mbps	1450	
	WiFi IEEE802.11 (ad hoc)	Sending @ 700 kB/s	1629	
		Receiving	1375	
		Idle	979	
	2G	Downloading @ 44Kbps	500	
		Handover 2G->3G	1389	2.4
	3G	Downloading @ 1Mbps	1400	
		Handover 2G->3G	591	2.5

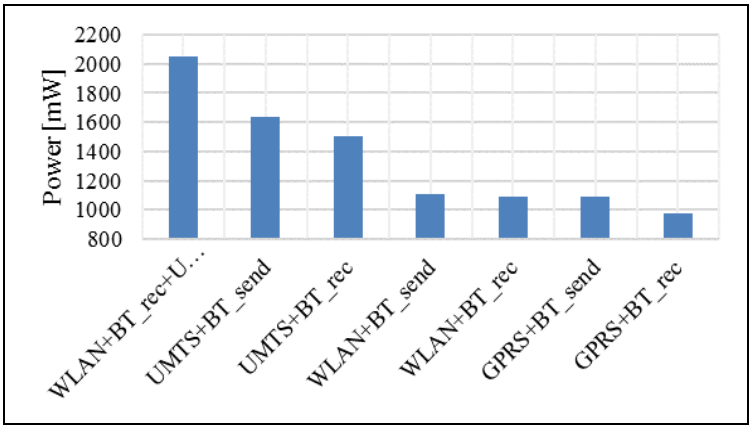


Fig. 2. Power Utilization per technology [6]

Another team of researchers [7] made an approach to model the energy consumption for Android smartphones, specifically the wireless interfaces of the device. They demonstrated the device capabilities from the power point of view with respect to time or to the amount of data transferred. Both aspects were proof that a node can handle a relatively high load suitable for practical implementations. In ad hoc networks: nodes would operate approximately four to five hours dealing with data transfer on WiFi. WiFi transferred 5.91 GB in download and 5.66 GB in upload.

3 Case Studies of Practical Implementations

From the preceding section, it is clear that advancements in hardware in the last few years can support modern applications processing requirements and maintain the battery life for hours. This section demonstrates the impact of this on real life MANET implementations and projects that benefit from modern mobile devices. Section A presents an under developing mobile telephony platform based on ad hoc network, whereas section B displays a commercial company's MANET based products.

3.1 Practical Wireless Ad Hoc Mobile Telecommunications

The Serval Project in [8], we believe it is the first practical mesh mobile telephony platform, is an example of MANETs real life implementation. It was initiated to provide mobile telecommunications for those who have abnormal events, such as war or terror attack, natural disaster or when governments deny their own citizen's mobile communications. These events lead to disconnect people form current mobile infrastructure oriented approach. Another situation for implementing such a project is for population in nomadic and remote locations who are not served well. Mobile companies may not invest in infrastructure implementation, as it is not economically feasible.

The system, which could operate in these circumstances, should be use free, not licensed – spectrum (WiFi), operate on WiFi enabled cell phones (as the only network hardware component), relay calls without a carrier, without telephone numbers allocation from authority and is completely self-organizing.

The Serval Project uses BATMAN [9] as the underlying mesh routing protocol. Distributed Numbering Architecture (DNA) was introduced to overcome telephone numbers allocation. It allows any device to request/respond the numbers from/to its neighboring devices. As described in the project, the telephone numbers self-allocation and distribution form untrusted environment. All introduced approaches that proposed to overcome this issue depend on the person who uses the system not the system itself. We think that this is not enough especially in such abnormal situations, we discussed before, in which this type of communication operates. The voice application consists of an embedded open source PBX software suite Asterisk

The System was tested in a three simulation cases: (a) Rescue mission, providing coverage to several square kilometers to be able to contact unreal lost person. (b) Provide service to quarantined remote group without any additional infrastructure

except their cell phones. (c) Reestablish telephony service for remote community. For the third test, they provided mobile telephone service for the first time to a village in a matter of 20 minutes. In addition, they delivered an alternate landline service to remote administration building from the open space around the village.

All three use cases were simulated by a fly-in-fly-out team in less than eight hours. The mesh telephony function was tested without support from any infrastructure using several HTC Dream Android phones; this is shown in Table 4. It was estimated that the effective range between phones was of the order of 500m, and likely>1km from ridge to ridge where the phones would have enjoyed clear Fresnel Zones.

Mobile phones can run for approximately 6-10 hours on the mesh depending on how much they are used. This is considered as an appropriate time to access these types of networks. One of the teams [8] is working on gathering statistics about battery life in different mobile phones.

Table 4. Specifications of Mobiles used in Practical Implementations [2]

		HTC Dream	Motorola Es400
	Announced	2009, February	2010, June
Features	OS	Android OS, v1.6 (Donut)	Microsoft Windows Mobile 6.5.3 Pro.
	Chipset	Qualcomm MSM7201A	Qualcomm MSM7627
	CPU	528 MHz ARM 11	600 MHz ARM 11
	GPU	Adreno 130	Adreno 200
	GPS	Yes	Yes, with A-GPS support
Connectivity	2G Network	GSM 850/ 900/1800 /1900	GSM 850/ 900/ 1800/ 1900
	3G Network	HSDPA 2100	HSDPA 850/ 1900/ 2100
	WLAN	WiFi 802.11 b/g	WiFi 802.11 a/b/g
	Bluetooth	Yes v2.0 with A2DP, headset support only	Yes, v2.0 with A2DP
Battery		Li-Ion 1150 mAh battery	Li-Ion 1540 mAh battery
	Stand-by	Up to 406 h	Up to 250 h
	Talk time	Up to 5 h 20 min	Up to 6 h

3.2 COCO Communication

CoCo Communications Corp. (CoCo) [10] is a US software company. They develop and deploy MANET solutions to provide reliable, secure, and scalable communications solutions for mobile and fixed environments. They have many software/hardware products, the next lines focus on one of them.

“CoCo Node” software, which is federally tested by the U.S. Coast Guard, U.S. Army and U.S. Navy, could be installed on a variety of mobile phones, Windows and Linux systems. It creates instant networks that do not depend on centralized infrastructure. Devices share their network connectivity with the rest of the network automatically. Devices are protected by certificate based security, which secure network communications on the network level, not the application layer. This protects the network against man in the middle and other attacks.

CoCo has its own proprietary modified distance vector protocol intended to increase usability, reliability, mobility, and security shown in Fig. 3 [10]. CoCo stack fits between existing OSI layer 2 and layer 3. It is divided into four layers: Routing, Circuit, Identity, and Addressing. Motorola ES400, this is shown in Table 4, is one of CoCo selling product that is powered by CoCo node software

Address Translation			
Identity Management			
Circuit Routing			
Packet Routing			
Cluster MANET	Satellite Data	Carrier Data	WiFi Hotspot

Fig. 3. CoCo Protocol Conceptual Layers [10]

4 Impacts on MANETS Developments

In this section, we discuss the impacts of hardware advancements on routing, quality of service and security. They are presented in sections 1, 2, and 3 respectively.

4.1 Routing

As we can see from [11], "due to small physical size, nodes in ad hoc networks have various constraints on bandwidth, memory, power and computational ability. Nodes usually have limited power sources which deplete very quickly with time and need to be recharged." This study has been done in 2006. Hence as seen in Table 1, the specifications of recent mobile devices, manufactured in 2012, are very highly developed compared to their counterparts from five years (in 2007) with the same manufacturers. Accordingly, we can elaborate here that processing and batteries capabilities should no longer considered a big concern for efficient routing protocols. Furthermore, recent studies in routing are based "Global Positioning System", GPS equipped devices. The proposed routing protocol called LANDY in [12]. LANDY is an acronym that stands for Local Area Network Dynamic. It is a position based routing protocol.

4.2 Quality of Service

Speaking of normal development path for MANETs, they should support real time communications such as audio, video, or even online games. This requires certain level of Quality of Service (QoS). QoS is defined as a set of bounds such as latency, jitter, throughput, and packet loss to be maintained by the network for a particular data flow [13]. Batteries and processing capabilities were always major concerns in MANETs. Utilizing the new hardware capabilities to enhance MANETs QoS will have positive impacts on many applications such as phone calls, the practical implementation shown in section 3.

4.3 Security

Nodes in MANETs are very susceptible to numerous attacks due to dynamic topology changes and open air medium. Hence, we can see that Intrusion Detection Systems (IDS) play an essential role in MANETs to secure the communication and dismiss malicious nodes [14]. As in [15], IDS architectures can be categorized into three: (a) standalone, (b) cooperative, and (c) hierarchical. The presence of IDSs brings a burden of processing and calculations that might impact the overall performance of MANETs. Therefore, all previous researchers were very conservative in implementing IDSs. They did not want to overload nodes with IDS processing. On contrary in recent days, with these remarkable advancements in processing and batteries, we believe that those IDS solutions should be revisited to increase their efficiency and accuracy. The hardware advancements opened the door also for Cryptography. Cryptography is also highly impacted; this can be seen in a recent study in [16], were the researchers used a powerful device with Core 2 Duo T7250, CPU and 3-GB RAM [16]. Hence, they used RSA [17], a public cryptography algorithm; and DSA [18], a digital signature algorithm. These two algorithms are quite known with complex computation complexity.

5 Conclusions and Future Work

Mobile devices are considered the cornerstone for MANETs developments. Throughout this paper, we presented the tremendous advancements in mobile phones capabilities, such as processing power and batteries developments. We also considered specifications of modern mobile phones that belong to different mobile phone manufacturers. They are capable of building MANETs and running modern applications. Case studies have shown that batteries could last for six hours on average during real life applications. It is concluded that we should not be worried about node capabilities and power consumption when developing different solutions for MANET. Furthermore, it is noteworthy to revisit all previous solutions that have been implemented for mobile devices with low capabilities with their limited energy; as these limitations should not resemble a concern any longer. Our future work will be a more detailed study on the impact of these advancements on MANET routing algorithms, QoS and security. A special attention will be given to specially location based routing as almost all manufactured devices come out with GPS devices enabled.

References

1. Kwang-Ting, C., Yi-Chu, W.: Using mobile GPU for general-purpose computing; a case study of face recognition on smartphones. In: International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, pp. 91–97 (2011)
2. Collective information gathered from the manufacture's websites (2013) and other tech sites, `http://www.apple.com`, `http://www.samsug.com`, `http://www.htc.com`, `http://www.gsmarena.com` and `http://www.thephonedatabase.com` (accessed March 15, 2013)

3. Shah, A.: Samsung laying groundwork for server chips, Analysts Say (2012), http://www.computerworld.com.au/article/441253/samsung_laying_groundwork_server_chips_analysts_say/ (accessed March 25, 2013)
4. Moore, G.E.: Cramming More Components Onto Integrated Circuits. In: Proceedings of the IEEE Solid-State Circuits Society Newsletter, vol. 86, pp. 82–85 (1998)
5. Nguyen, V., Hao, S., Szajman, J.: WiiKey: An Innovative Smartphone Based Wi-Fi Application. In: International Multisymposiums Computer and Computational Sciences, IMSCCS 2008, Shanghai, pp. 91–97 (2008)
6. Perrucci, G.P., Fitzek, F.H.P., Widmer, J.: Survey on Energy Consumption Entities on the Smartphone Platform. In: IEEE 73rd Vehicular Technology Conference (VTC Spring), Yokohama, vol. 15, pp. 5–20 (2011)
7. Kalic, G., Bojic, I., Kusek, M.: Energy consumption in android phones when using wireless communication technologies. In: Proceedings of the 35th International Convention MIPRO, Opatija, pp. 754–759 (2012)
8. Gardner-Stephen, P.: Serval Project Developers (2010), http://developer.servalproject.org/site/ (accessed April 10 , 2013)
9. Johnson, D., Ntlatlapa, N., Aichele, C.: A simple pragmatic approach to mesh routing using BATMAN. In: 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa (2008)
10. (CoCo), C.C.C., Next Generation Networking (2012), http://www.cococorp.com (accessed April 30, 2013)
11. Safdar, G.A., McGrath, C., McLoone, M.: Limitations of existing wireless networks authentication and key management techniques for MANETs. In: International Symposium Computer Networks, Istanbul, pp. 101–105 (2006)
12. Macintosh, A., et al.: Local Area Network Dynamic (LANDY) routing protocol: A position based routing protocol for MANET. In: 18th European Wireless Conference, European Wireless, EW 2012, Poznan, Poland, pp. 1–9 (2012)
13. Crawley, E., et al.: A framework for QoS-based routing in the internet, RFC (1998)
14. Mishra, A., Nadkarni, K., Patcha, A.: Intrusion detection in wireless ad hoc networks. IEEE Wireless Communications 11, 48–60 (2004)
15. Panos, C., Xenakis, C., Stavrakakis, I.: A novel Intrusion Detection System for MANETs. In: Proceedings of International Conference on Security and Cryptography, SECRYPT, Athens, pp. 1–10 (2010)
16. Shakshuki, E.M., Nan, K., Sheltami, T.R.: A Secure Intrusion-Detection System for MANETs. IEEE Transactions on Industrial Electronics 60, 1089–1098 (2013)
17. Selby, A., Mitchell, C.: Algorithms for software implementations of RSA. In: IEEE Proceedings Computers and Digital Techniques, vol. 136, pp. 166–170 (1989)
18. Seys, S., Preneel, B.: Power consumption evaluation of efficient digital signature schemes for low power devices. In: IEEE International Conference Wireless and Mobile Computing, Networking and Communications, WiMob 2005, vol. 1, pp. 79–86 (2005)

A Virtualized Network Testbed for Zero-Day Worm Analysis and Countermeasure Testing

Khurram Shahzad, Steve Woodhead, and Panos Bakalis

Internet Security Research Laboratory, University of Greenwich, Chatham, Kent, UK
{K.Shahzad,S.R.Woodhead,P.Bakalis}@gre.ac.uk

Abstract. Computer network worms are one of the most significant malware threats and have gained wide attention due to their increased virulence, speed and sophistication in successive Internet-wide outbreaks. In order to detect and defend against network worms, a safe and convenient environment is required to closely observe their infection and propagation behaviour. The same facility can also be employed in testing candidate worm countermeasures. This paper presents the design, implementation and commissioning of a novel virtualized malware testing environment, based on virtualization technologies provided by VMware and open source software. The novelty of this environment is its scalability of running virtualised hosts, high fidelity, confinement, realistic traffic generation, and efficient log file creation. This paper also presents the results of an experiment involving the launch of a Slammer-like worm on the testbed to show its propagation behaviour.

Keywords: Worms, malware, Slammer, testbed, virtualization, VMware.

1 Introduction

Computer worms are a serious potential threat to network security. The high rate of propagation of worms and their ability to self-replicate make them highly infectious. A zero-day worm is a type of worm that uses a zero-day exploit; a publically unknown and un-patched vulnerability in network daemon software [1]. SQL Slammer is considered to be the fastest zero-day random scanning worm in history as it infected more than 75K hosts in less than 10 minutes [2]. The Stuxnet worm is a recent addition to this class of malware that spies on and subverts supervisory control and data acquisition (SCADA) systems and was the first network worm to include a programmable logic controller (PLC) rootkit [3].

Whilst other experimental malware testbeds have been reported, further improvements in this area will allow greater effort to be exerted in the development of malware defense techniques, such as worm countermeasures. Physical network setup [4, 5, 6, 7], simulation [8, 9, 10, 11], emulation [12, 13, 14] and virtualization [15, 16, 17, 18, 19, 20, 21] are some of key techniques previously reported for creating such experimental testbeds. The major challenges in implementing such a test environment are fidelity, scalability, confinement, realistic benign and malicious traffic generation,

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 54–64, 2013.

efficient log file creation, rebuild and configuration time, analysis and visualization, multi platform support, portability to different physically distributed testing environments, and flexibility in adjusting to experimental needs [17]. These diverse requirements of network and security experimental research are not well met by any single existing testbed. Competing methods remain popular because each tries to cover some portion of these requirements. Hence there is a need to design, implement and evaluate a novel virtual testing environment which incorporates increased granularity and instrumentation functionality.

With the aim of addressing these points, this paper presents the design, implementation and commissioning of a novel virtualized malware testbed, which employs VMware virtualisation technology and a range of open source software. We refer to the testbed as the Virtualized Malware Testbed (VMT). The novelty of this environment is its scalability, high fidelity, confinement, realistic traffic generation, and efficient log file creation. The paper also presents the results of an experiment involving the launch of a Slammer-like worm within VMT, to show the propagation behavior of the worm, and to validate the operation of the testbed.

The remainder of paper is presented as follows: Section 2 summarises the relevant previous work; Section 3 details the design, implementation and commissioning of VMT; Section 4 presents the experimental methodology and results of launching the Slammer-like pseudo-worm; and finally Section 5 concludes the paper with a discussion summarizing the findings and identifying any limitations, as well as summarising potential future work in this area.

2 Relevant Previous Work

Various network and malware testing environments have been built and proposed in the past which can be classified into the following categories:

- Physical machine testbeds
- Simulation testbeds
- Emulation testbeds
- Virtual machine testbeds
- Full system virtualization testbeds

2.1 Physical Machine Testbeds

Physical machine testbeds employ real physical hosts and network hardware for conducting research experiments. Emulab [4] was a distributed physical network setup, implemented for conducting research experiments. It consists of 218 physical nodes distributed between two US universities. Netbed [4] is a simulation environment implemented on Emulab that provides time and space sharing and employs ns-2 [11] for research and development. Emulab evolved into DETER [5], which is a cluster based

testbed, consisting of high end workstations and a control software. It uses high-performance VLAN-capable switches to dynamically create nearly arbitrary topologies among the nodes. It was the first testbed to be remotely accessible through the public internet infrastructure. The 1998 DARPA off-line intrusion detection evaluation [6] and LARIAT [7] are also two physical machine testbeds sponsored by US Air Force and developed at the Lincoln Laboratory, MIT.

2.2 Simulation Testbeds

Simulation testbeds employ simulation tools to conduct network experiments. PDNS and GTNetS [8] were two network simulators for developing packet level worm models. These simulators allow an arbitrary subject network configuration to be specified consisting of scan rate, topology and background traffic. On the basis of defined input parameters, various types of outputs such as number of infected hosts in any given instance, sub-millisecond granularity of network event statistics or a global snapshot of the entire system are produced. Ediger reported the development of the Network Worm Simulator (NWS) [9], which implements a finite sate machine concept to simulate network worm behavior. Tidy et al [10] have reported a large scale network worm simulator aimed at the investigation of fast scanning network worms and candidate countermeasures.

2.3 Emulation Testbeds

Emulation testbeds provide a compromise between simulation and real world testing. ModelNet [12] and PlanetLab [13] are two emulated testbeds, implemented for general networking and distributed system experiments. In ModelNet, unmodified applications run on edge nodes, configured to route all their packets through a scalable core dedicated server cluster, by emulating the characteristics of a special target topology. PlanetLab was developed for the purpose of creating world-wide distributed systems, and has a dual nature of being used by developers and clients. Honeypots such as Honeyd [14] can also be classified as an emulation system as it has been used in many recent security systems for malware detection and capture.

2.4 Virtual Machines Testbeds

Virtual machine testbeds employ virtualization technologies as their main building block to conduct security and network experiments. ReVirt [15] is an advanced VM-based forensic platform which enhances individual virtual machines with efficient logging and replay capabilities, by redirecting log files from the guest OS to the host OS, for intrusion analysis purposes, thereby making it possible for malware analysis researchers to replay the malware exploitation process in an intrusion by intrusion fashion. Based on ReVirt [15] research, another platform VMWatcher [16] was

developed that places the anti-virus system in the hypervisor layer, in order to be unreachable by the attacker. Research in SINTEFF ICT [17] has examined the effect of malicious software on a Windows XP workstation by utilizing Nessus [22] as the attacker, Wireshark [23] as a sniffer, Snort [24] as a NIDS and Sysinternals[25] to provide HIDS functionality.

2.5 Full System Virtualization Testbeds

Full system virtualization testbeds employ full virtualization; a technique that provides a type of virtual machine environment with complete simulation of the underlying hardware. vGround [18] has extended UML's virtual networking capabilities by supporting a VM-create-VM approach to automatically extend the network size. It uses Snort [24] and Bro [26] as NIDS and Kernort [27] as a HIDS to monitor worm target discovery and propagation. ViSe [19] provides a virtualization platform where malware exploits can be tested against the entire range of x86 based operating systems under controlled conditions, while being monitored by a NIDS. V-NetLab [20] has implemented a model based on DETER's [5] remote access capability by utilizing data link layer virtualization and packet encapsulation, thereby providing a more secure means of remote access to security related Testbeds. Golath [21] is a virtual network based on a Java Virtual Machine (JVM) and the Ultra light-weight abstraction level (ULAL). It provides a virtual environment to run any application written in Java, independent of the type of host operating system. System behavior can be monitored in this environment by adding different Java plug-in extensions.

2.6 Motivation

As far as the authors of this paper are aware, no previous virtualized malware testing environment has provided a scalable solution with a large number of virtual machines for security experiments by using VMware technologies. Isolation of the test environment from the management network with remote access also seems to be a problem. It is also noted that no previous reported work has produced infection and propagation analysis of any fast random scanning worm such as SQL Slammer in a real network with real world slammer exploitable conditions.

3 Virtualized Malware Testbed (VMT) Design and Capabilities

The Virtualized Malware Testbed (VMT) was designed with the intention of employing it in an investigation of exiting and hypothetical zero-day worms; and testing candidate worm countermeasures. Our goals of implementing VMT were experimental scalability, fidelity, repeatability, programmability, remote access and efficient log file creation.

3.1 Architecture, Design and Implementation

VMT uses VMware ESXi [28] as the core virtualization technology and Damn Small Linux (DSL) [29] as the main virtualised operating system. It also uses Quagga [30] to provide a software routing suite. VMware vCenter Server [31] provides a graphical user interface to manage VMware ESXi servers remotely.

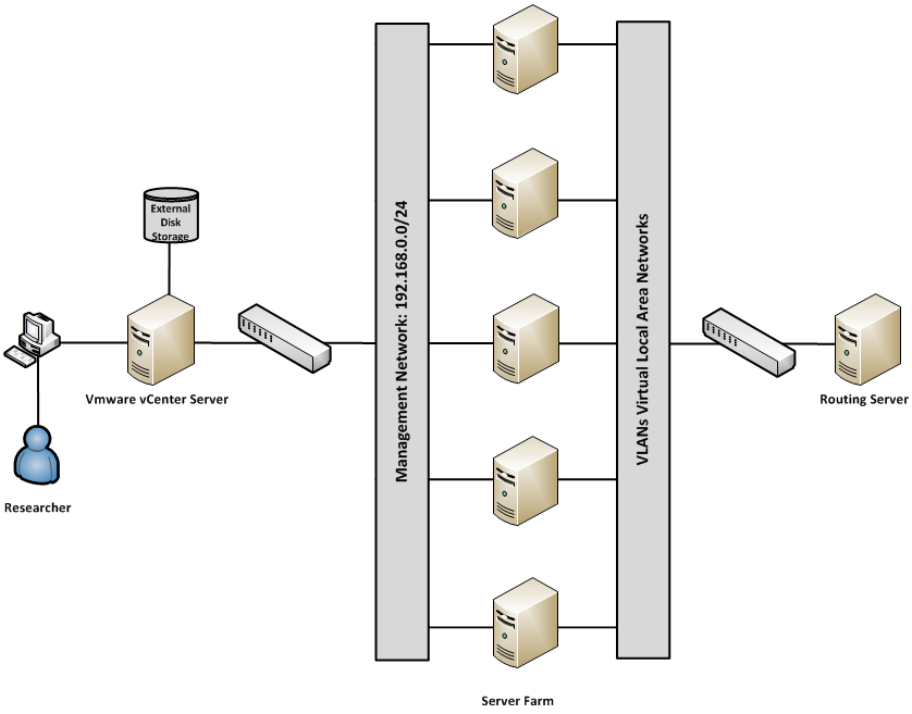


Fig. 1. Physical Network Setup

Figure 1 illustrates the physical architecture of VMT. It consists of a server farm with five servers, a management server and Ethernet switches. Each server in the server farm is running ESXi while the management server is running VMWare vCenter Server. One network interface card in each server farm machine is connected to a logically isolated management network along with the management server; thereby allowing access to all resources from one interface. Multiple virtual topologies can be created within the server farm by using virtual local area networks (VLANs) and Quagga. Each DSL virtual machine image is installed with 32 MB of memory and 1 GB hard disk. Table 1 summarizes the hardware and operating systems which make up the VMT infrastructure.

Table 1. VMT Hardware and Operating System Infrastructure

	Processors	**No of cores**	**Operating System**	**Memory**	**Storage**	**VMs**
Server 1	i7	6	ESXi 5.1	64 GB	1 TB	DSL, Ubuntu
Server 2	i7	4	ESXi 4.1	24 GB	1 TB	DSL, Ubuntu
Server 3	i7	4	ESXi 4.1	24 GB	1 TB	DSL, Ubuntu
Server 4	Xeon	4	ESXi 4.1	8 GB	512GB	DSL, Ubuntu
Server 5	Xeon	4	ESXi 5.1	8 GB	512GB	DSL, Ubuntu
Management Server	i7	4	Windows Server 2003 R2	8 GB	2 TB	N.A
Routing Server	i5	2	Ubuntu Quagga	4 GB	512GB	N.A

A minimum rebuild and configuration time are key goals of any security testing environment. VMware vCenter Server provides PowerCLI [32]; a command line interface tool that allows administrators to create simple and robust scripts to automate the main tasks, including virtual machines cloning.

4 Experimentation

4.1 Slammer-Like Pseudo Worm

In order to analyze the behavior of a SQL Slammer-like worm; we developed a network daemon which implements a Slammer-like pseudo-worm. This daemon listens on UDP port 1434 and upon receiving a datagram with an appropriate authentication string (included for safety reasons), it begins generating UDP datagrams addressed to port 1434 and to random IP addresses. The speed of datagram generation per second, and the pool from which the random destination IP addresses are chosen are configurable parameters. We have also implemented a logging server. At the point of "infection", the pseudo-worm daemon sends an infected time message to the central logging server.

4.2 Experimental Setup

We have setup a virtual test network comprising of a single Class A address space 10.0.0.0/8 but divided into four subnets; 10.0.0.0/10, 10.64.0.0/10, 10.128.0.0/10 and 10.192.0.0/10 as shown in Figure 2. These four subnets are connected through a central router by using RIP, configured on Quagga. Four further Quagga based routers

are implemented (one for each subnet). One Linux based virtual machine is running in each subnet to provide a DHCP service. DSL is installed with the pseudo-worm daemon on each of the susceptible virtualised hosts. All hosts in the network are time synchronized by using the Network Time Protocol (NTP).

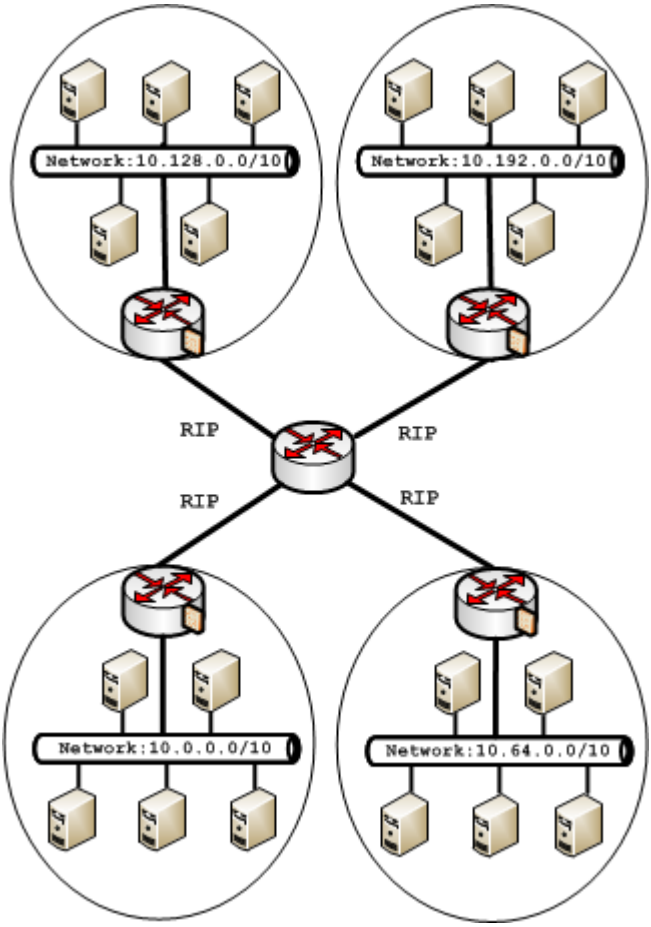


Fig. 2. Virtual Experimental setup for Slammer- like Pseudo Worm Behavior Analysis

4.3 Experimental Methodology

Moore et al. [2] reported a set of key characteristics of the Slammer worm outbreak in 2003 and these were used to set up the experimental parameters. Moore et al. reported that 18 hosts per million of the entire IPv4 addresses space were susceptible to infection. They also observed that the Slammer worm exhibited an average scan rate of 4,000 datagrams per infected host per second.

A single class A network has 2^{24} hosts, and so will contain $2^{24} * 0.000018 = 302$ susceptible hosts. On this basis, 302 virtual machines with the Slammer like pseudo-worm

daemon were deployed across the four subnets. Each worm daemon was configured to scan within a single class A network (10.0.0.0/8). In order to avoid overloading the server farm hardware (in which case we would have been measuring the effect of the hardware restrictions, rather than the properties of the worm) we scaled back the average worm scanning rate by a factor of 80. Therefore, based on an average scan rate reported by Moore et al of 4000 scans per second, we configured the Slammer-like network daemons to scan at 50 scans per seconds in our experiment.

4.4 Experimental Results

Figure 3 shows the results of the experiment, with the time axis scaled down by a factor of 80, to make the results comparable with the real infection event of 2003, reported by Moore et al [2].

In order to provide a baseline comparison, we have also plotted a susceptible/infected analytical model, based on the Logistic Equation, reported by Ediger [9].

The VMT experiment achieved infection of 99% of vulnerable hosts within approximately 14 minutes. This time is directly comparable with that reported in [2] for the real Slammer event of 2003. We have also plotted available data from [2] for the 2003 event, in Figure 3 (empirical data is only available for the first 4 minutes of infection), and it can be seen that the VMT experimental results are again, broadly comparable.

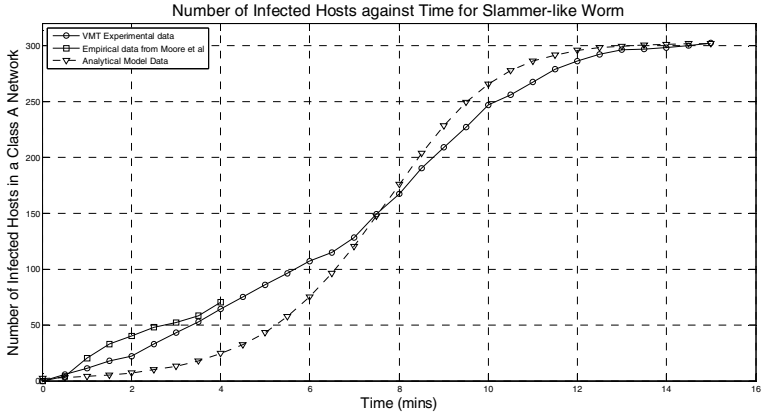


Fig. 3. Experimental Results for Slammer-Like Worm Infection on VMT

5 Discussion

The cyber-epidemiological analysis of zero-day internet worms remains a significant challenge and use of virtualized testbeds remains a viable tool for such research. This paper has presented a novel Virtualized Malware Testbed (VMT) for worm testing based on VMware ESXi and open source software. We have also demonstrated its feasibility for epidemiological experimentation for a Slammer–like pseudo-worm.

In comparison with other network and security testing environments, VMT provides an effective, scalable, remotely manageable and isolated environment, which also incorporates efficient log creation. It is expected that VMT will be a useful experimentation environment for epidemiological investigations of existing and hypothetical zero-day worms, as well as the investigation and evaluation of candidate countermeasures.

5.1 Limitations and Future Work

This paper has reported the design, implementation and initial testing of VMT with a single network worm type. The experimentation has also been limited to the scale of a single class A network (circa 16M hosts).

In terms of future work, we shall be exploring the use of VMT to explore the stochastic properties of worms, as well as its ability to investigate other types of network worm. We also expect to experiment with a range of candidate worm countermeasures, and to explore the applicability of VMT for charactering the epidemiology of more sophisticated malware threats, such as Stuxnet.

Acknowledgment. VMware ESXi, VMware VCenter servers are provided as part of VMware Academic Program. (http://www.vmware.com/partners/academic/program-overview.html).

References

1. Weaver, N., Paxson, V., Staniford, S., Cunningham, R.: A taxonomy of computer worms. In: Proceedings of 2003 ACM Workshop on Rapid Malcode, pp. 11–18. ACM Press, New York (2003)
2. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., Weaver, N.: Inside the Slammer worm. IEEE Security and Privacy 1(4), 33–39 (2003)
3. Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy 9(3), 49–51 (2011)
4. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S.: An integrated experimental environment for distributed systems and networks. In: Proceedings of 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, pp. 265–270. USENIX (2002)
5. Benzel, T., Braden, R., Kim, D., Neuman, C.: Design, deployment and use of the DETER testbed. In: Proceedings of DETER Community Workshop on Cyber Security Experimentation and Test 2007, Berkeley, CA, USA, pp. 1–8. USENIX (2007)
6. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K., Zissman, M.A.: Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX), vol. 2, pp. 12–26. IEEE Press, New York (2000)
7. Rossey, L.M., Cunningham, R.K., Fried, D.J., Rabek, J.C., Lippmann, R.P.: LARIAT: Lincoln Adaptable Real Time Information Assurance Testbed. In: Proceedings of IEEE Aerospace Conference, Big Sky, Montana, USA, vol. 6, pp. 2671–2682. IEEE (2002)

8. Perumalla, K.S., Sundaragopalan, S.: High fidelity modeling of computer network worms. In: Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC), Tucson, AZ, USA, pp. 126–135. ACSA (2004)
9. Ediger, B.: Simulating Network Worms, http://www.stratigery.com/nws/
10. Tidy, L., Woodhead, S.R., Wetherall, J.C.: A Large-scale Zero-day Worm Simulator for Cyber-Epidemiological Analysis. UACEE International Journal of Advances in Computer Networks and Security 3(2), 69–73 (2013)
11. ns (network simulator), http://www.isi.edu/nsnam/ns
12. Vahdat, A., Yocum, K., Walsh, K., Mahadevan, P.: Scalability and accuracy in a large-scale network emulator. In: Proceedings of USENIX 5th Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, USA, pp. 271–284. USENIX (2002)
13. Peterson, L., Anderson, T., Culler, D., Roscoe, T.: A blue print for introducing disruptive technology into the internet. SIGCOMM Computer Communication Review 33(1), 59–64 (2003)
14. Provos, N.: A virtual Honeypot framework. In: Proceeding of USENIX 13th Security Symposium, San Diego, USA, pp. 1–14. USENIX (2004)
15. Dunlap, G., King, S., Cinar, S., Basrai, M., Chen, P.: ReVirt: enabling intrusion analysis through virtual machine logging and replay. In: Proceeding of USENIX 5th Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, pp. 208–223. USENIX (2002)
16. Jiang, X., Wang, X.: Stealthy malware detection through VMM-Based "out-of-the-box" semantic view reconstruction. In: Proceedings of 14th ACM Conference on Computer and Communication Society (CCS), Alexandria, VA, USA, pp. 128–138. ACM (2007)
17. Jenson, J.: A novel testbed for detection of malicious software functionality. In: Proceeding of Third International Conference on Availability, Security and Reliability, Barcelona, Spain, pp. 292–301. IEEE (2008)
18. Jiang, X., Xu, D., Wang, H.J., Spafford, E.H.: Virtual Playgrounds for Worm Behavior Investigation. In: Valdes, A., Zamboni, D. (eds.) RAID 2005. LNCS, vol. 3858, pp. 1–21. Springer, Heidelberg (2006)
19. Årnes, A., Haas, P., Vigna, G., Kemmerer, R.A.: Digital Forensic Reconstruction and the Virtual Security Testbed ViSe. In: Büschkes, R., Laskov, P. (eds.) DIMVA 2006. LNCS, vol. 4064, pp. 144–163. Springer, Heidelberg (2006)
20. Sun, W., Katta, V., Krishna, K., Sekar, R.: V-netlab: an approach for realizing logically isolated networks for security experiments. In: CSET 2008: Proceedings of the Conference on Cyber Security Experimentation and Test, Berkeley, CA, USA, pp. 1–6. USENIX (2008)
21. Fagen, W., Cangussu, J., Dantu, R.: A virtual environment for network testing. Journal of Network and Computer Applications Archive 32(1), 184–214 (2009)
22. Nessus Vulnerability Scanner, http://www.tenable.com/products/nessus
23. Wireshark, http://www.wireshark.org/
24. Snort, http://www.snort.org/
25. Windows Sysinternals, http://technet.microsoft.com/en-US/sysinternals
26. The Bro Network Security Monitor, http://www.bro.org/
27. Jiang, X., Xu, D., Eigenmann, R.: Protection mechanisms for application service hosting platforms. In: Proceedings of 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2004), Chicago, Illinois, USA, pp. 633–639. IEEE Computer Society (2004)

28. VMware ESXi, http://www.vmware.com/products/vsphere/esxi-and-esx/overview.html
29. Damn Small Linux (DSL), http://www.damnsmalllinux.org
30. Quagga Software Routing Suite, http://www.nongnu.org/quagga
31. VMware vCenter Server, http://www.vmware.com/products/vcenter-server/overview.html
32. VMware vSphere PowerCLI, http://communities.vmware.com/community/vmtn/server/vsphere/automationtools/powercli?view=overview

A Categorized Trust-Based Message Reporting Scheme for VANETs

Merrihan Monir[1], Ayman Abdel-Hamid[1], and Mohammed Abd El Aziz[2]

[1] College of Computing and Information Technology, Arab Academy for Science, Technology, and Maritime Transport, Alexandria, Egypt
merrihan@hotmail.com, hamid@aast.edu
[2] Information Systems Department, Faculty of Computers and Information Sciences, Ain Shams University, Cairo, Egypt
mhashem100@yahoo.com

Abstract. In Vehicular Ad Hoc Networks (VANETs), trust establishment among communicating vehicles is important to be established to secure messages' exchange and reliability. In this paper, a categorized decentralized trust management and evaluation scheme for nodes in VANETs is presented. Role-based trust and experience-based trust is integrated, while using an opinion piggybacking process when needed. Each node is evaluated individually according to its interactions during event reporting. Based on this evaluation, a node is assigned a category level (according to its trust value) and a confidence measure that determine the degree of trustworthiness of a node's generated reports. The scheme integrates role-based trust with experience-based trust, to form a combined trust scheme, taking into account the history of the driver's interactions with other vehicles. Case studies, scheme analysis and validation, demonstrate early malicious node detection, which leads to an efficient reporting scheme.

Keywords: Advisory network, efficient reporting system, malicious node detection, message verification and broadcasting, penalty system, trust management in VANET.

1 Introduction

Traffic congestions, road accidents and maintenance, are the major problems of current traffic systems, in which drivers should be fully aware of the surrounding environment. A Vehicular ad hoc network (VANET) is a class of ad-hoc networks. In VANETs, there are two types of communication [1], vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) also known as vehicle-to-roadside unit (V2R). In V2V communication, vehicles send and receive messages to and from one another. These messages can be reporting road congestion, accidents ahead, etc., known as safety messages. V2I communication occurs between nodes and road side infrastructure, e.g., reporting an event or a malicious node, finding nearest gas station, or online toll payment, among others. Vehicular communications consists of vehicles (nodes), road

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 65–83, 2013.

side units (RSUs) and governmental transportation authorities (GTA). An RSU is used for broadcasting emergency road-safety messages, changed road-condition notifications, or locality information. GTA is the governmental transportation authority, responsible for driver licensing, vehicle registration to the system, storing vehicle's information, issuing vehicles and infrastructures cryptographic credentials used for V2V and V2I communications.

VANETs are ephemeral networks [2], where connections between nodes are short lived. The density of the network changes continuously, e.g. it is higher during rush hours and lower at night time. Since nodes keep moving in and out of communication range, two communicating nodes might not interact in future. Therefore, node evaluation should be performed in a decentralized way by a trusted authority. Processing devices and secure storage medium are installed in vehicles to allow complex calculations during trust evaluation to take place.

It is important for vehicular ad hoc environments to ensure traffic safety, by delivering the right information to drivers in a time-sensitive manner. This is not always easy due to the presence of malicious or greedy nodes, where false information could be broadcasted misleading other nodes in the scene. Thus, establishing trust for nodes is an essential factor in order to determine whether their claimed transmitted information is reliable [3].

In this paper, a node trust management framework is presented based on a decentralized evaluation scheme for vehicles and drivers. Each driver license Id number (node) is concatenated to the vehicle Id number, before sending a message, only revealed when needed by authorized entities. Trust evaluation is performed in a decentralized fashion by a RSU, which is deployed in different administrative regions. Driver Id no. is assigned a concrete trust value and a confidence measure. The trust value reflects node historical interactions through its driving life time and to which extent this node is considered trusted. The confidence measure is a reflection of the trust value that proves node's degree of trustworthiness. Based on this trust value, each node will be placed in a specific category that reflects its behaviour and extent of trustworthiness of its generated reports, and whether or not to believe it in the future. A node's category could be revealed by recipient nodes to provide decision making support [4]. In addition, this forms to the foundation of an advisory network for nodes and gives credit to real-time applications such as event reporting and crash reporting, which are time sensitive. Moreover, it allows vehicles to distinguish malicious nodes in real time. Therefore, many people's lives, processing time and effort could be saved.

The contributions of this paper can be summarized as follows; presentation of a node trust management framework for VANETs modelled in a categorized decentralized scheme, introducing a penalty system for detected malicious nodes, the foundation of an advisory network for V2I and V2V communications, and efficient message verification for broadcasting purposes by RSU.

The rest of this paper is organized as follows. Background and related work is presented in section 2. The proposed scheme is detailed in section 3. The proposed scheme's validation is outlined in section 4. Finally, section 5 concludes the paper and discusses future work.

2 Background and Related Work

Safety and non-safety applications maintain drivers and passengers requirements on roads. Such applications should be secured and ready to encounter different attacks initiated by malicious nodes. Security in VANETs will be presented in section 2.1, trust management in VANETs is listed in section 2.2 and related work is discussed in section 2.3.

2.1 Security in VANETs

Message authentication [5] and integrity [7] are important security requirements in VANET applications. Message authentication guarantees that the message comes from its original sender, signed by his own private key. Message integrity means the contents of the message as received by the receiver are the same as originally sent by the sender

Safety messages and correct event reporting play an important role in vehicular systems. Since, what would be the use of applying good authentication protocols and ensuring message integrity, whereas, message content is deceptive. Nodes will make their decisions according to what they read and understood in a report, e.g. in a critical situation, where an ambulance car will be searching for the shortest free path to go through, for the sake of saving lives. If a malicious node claimed that this path is jammed, the ambulance will change lanes, and a life of a person will be endangered. Thus, trust in message content and securing applications, together will lead up to an efficient reporting vehicular system.

To secure messages broadcasting, from greedy or malicious nodes, digital signatures and exchanged secret keys were proposed to help build trust among vehicles in V2V and V2I communications and to guarantee that messages had reached their intended destination. However, the presence of compromised nodes which routes false information can pass through such security protocols. This is because of the lack of available efficient schemes that could evaluate and verify the message content. Moreover, to the best of our knowledge, no scheme maintains records of previous interactions of a node for a long term, throughout its driving life time and in different regions. In the previous schemes [3], [4], [11], [12], [14], trust decisions were determined by evaluating exchanged security keys or by monitoring incoming messages and nodes' behaviour during short-lived sessions of communication.

Several security frameworks and trust schemes were proposed to secure the network from various security attacks [6], [7], [8]. However, false information injection, on and off attack, new comer attack, betrayal attack, sybil attack, collusion attacks, inconsistency attack and network jamming are the most common attacks in vehicular systems.

2.2 Trust Management in VANETs

"Trust" is the key element in creating a trusted vehicular environment which promotes security in vehicular networks. Trust is either in human behavior or in the deployed

hardware. The availability of both types of trust form a trusted communicating environment. Few trust schemes had been introduced to enforce honest information sharing between communicating nodes [7], [15]. Characteristics of trust schemes [15] in vehicular environments should be: decentralized, cope with scarcity of information, location and time specific, scalable and objective. Since a vehicle could have many drivers driving it, or a driver could be driving many vehicles (e.g. city cab), it is recommended that the trust value to be concatenated to the driver's driving license ID. This is mainly to guarantee that each person is judged according to their own behavior, even though driving different vehicles. Thus, encouraging honesty and discarding malicious or greedy drivers from reaching their unfair aims. It is recommended that nodes do not report events they didn't witness by them, in order not to be judged for false sent information. A history record should be available for each driver, to be able to judge each node independently. Trust should be evaluated at the authority level, to guarantee accuracy and fairness during computing and privacy and confidentiality for user nodes. Trust must be a global variable with specific range.

2.3 Related Work

Several efforts had been introduced in order to build trust between communicating nodes. Nevertheless, none of the previous protocols had fulfilled all the trust management requirements [7]. This was mainly because trust evaluation was done at the node level, which by time deletes all previous records it had for other nodes, since V2V communications are short-lived. Chaung et al. [11] state that the first mistrustful node becomes trustful and authenticated, once it obtains sufficient authorized parameters, so it can authorize other mistrustful nodes. If an adversary node was authenticated as trustful, it may misuse this trust gained to authorized and authenticate other misbehaving nodes. A user can also have more than one identity in the network.

Sumra et al. [12], state that if trusted node A communicates with node B safely, then node B becomes trusted and so on. Thus, it provides a chain of trust between a communicating group of nodes. But there will always be a risk where the first node communicating with the new comer node, will always be the victim. In addition, a malicious node can join a new group that has no idea about its bad history, and deceive nodes within this new group.

Sumra et al. [13] depend on a 16 digit secret code to ensure secure key renewal. The main drawback of this solution occurs at the entry point where client and service provider authentication task is performed. The channel could be congested when number of users increases, e.g. in a highway. Biswas et al. [14], state that if an emergency road-safety application message is generated by a trusted central authority, the issued message is broadcasted by RSUs to nodes on behalf of the originator of the message. This is a partially delegation of authorities. But this system is short-lived, because after the broadcasting task ends, it is not clear which nodes are trusted.

Huang et al. [2] discuss the problem of information cascading and oversampling, where it takes the majority voting for an event. Majority voting takes time in collecting other nodes opinions and taking a decision accordingly. Abumansoor et al. [16] discuss that if an obstacle was between two nodes that wish to communicate,

they can find an intermediate node to send through it the message. Unfortunately, this doesn't build any kind of direct trust between the 2 main nodes.

More reliable trust management schemes were introduced by Minhas et al. [3] and extended in [4]. It takes into account role-based trust and experience-based trust. Its main drawback is that too much calculations take place at the node level to evaluate the trust value of another node, and decide whether to adopt its opinion or not. These calculations are wasted because these couple of nodes have a very low chance to communicate again in future. This leads to time and processing consumption. In addition, certain variables are determined by each node, such as increment and decrement factors. Thus, trust values results may differ according to each node's assumption, whereas, the evaluated node is the same. Therefore, trust should be a public factor, to make efficient use of previous calculations, where also variables should have a clear specified value.

Chen et al. [10] present opinion piggybacking, where nodes add their own opinion to the forwarded message. This dramatically increases message size. In a high density area, many nodes will be forwarding the same message attached to it their opinion. This could lead to network congestion and node's memory high consumption.

3 Categorized Trust-Based Message Reporting Scheme for VANETs (CTMR)

Since drivers of the vehicles are human beings, it is assumed that human behavioural tendencies will be reflected in the behaviour of each node [2]. The scheme promotes the concept of "trust is hard to gain, but easy to lose".

The proposed scheme is based on the combination of role-based trust and experience-based trust, assuming direct trust, taking into account historical node interactions and traffic violations throughout its driving life time. The protocol uses opinion piggybacking method [10] in certain situations, upon request when needed. Trust computation is done at the governmental level, to guarantee privacy, security and efficiency. Section 3.1 outlines the reporting scheme objectives. Section 3.2 presents trust scheme basics. Section 3.3 outlines scheme design and trust evaluation steps. Section 3.4 presents the reliance on an advisory network concept. Section 3.5 discusses the adopted penalty system. Section 3.6 presents thoughts and discussions about the proposed scheme. Finally, section 3.7 compares the presented schemed versus related work and highlights its advantages.

3.1 Objectives

The main objectives of the proposed scheme could be summarized as follows:

- To counter the attacks previously mentioned in section 2.1.
- To minimize the number of calculations and processing time.
- To decrease memory consumption.

- To ensure accurate results.
- Efficient message verification for broadcasting purposes through the use of a confidence measure.
- Provide an efficient and fast trust evaluation scheme.
- Establishing a strong database containing contributing nodes' performance and history.
- Providing a categorized public scheme where each vehicle assigned a category reflecting its trustworthiness.
- Enabling nodes to take on-time accurate decisions upon receiving reports.
- Temporarily blockage for detected malicious node.
- Employing penalties on malicious nodes.
- Forming an advisory network of trusted nodes for opinion piggybacking purposes.
- System dynamics of contributing nodes.

3.2 Trust Scheme Basics

The protocol adopts the schemes introduced in [3], [4], in order to build a more reliable trust evaluation scheme for vehicles, while capturing the trust history of each node. The proposed scheme is based on the combination of role-based trust and experience-based trust, using opinion piggybacking upon request [10], assuming direct trust, taking into account historical node interactions and traffic violations throughout its driving life time. Trust computation is performed at the governmental level, to guarantee privacy, security and accuracy.

Assume that A_u is a node, which has three states either a beginner node (new driver with no past interactions), user nodes (driver who had been driving for some time) and terminated nodes (drivers who are no longer driving for some reason). Each state is declared so, only by the GTA. Assume the following scenario where node A_u sends message M_i to the nearest RSU within its communication range. M_i is evaluated by the RSU, if its contents are proven to be true under certain criteria, RSU broadcasts message M_i and adds trust credit to A_u and to nodes that helped in the message verification process. If its contents were proven to be false, RSU takes action accordingly and decreases trust value of A_u. Message verification and evaluation is done in a decentralized way by RSU, because as node moves in different paths and areas, it communicates with several different RSUs throughout its journey. Each RSU, sends in return its opinion credit of message M_i sent from node A_u (concatenated by its driver ID no.) to be stored in GTA, as shown in Fig. 1. Storing records of each node's interactions is done by the GTA in a centralized database. Each vehicle stores its personal data (driving ID no., vehicle Id no., public and private keys, etc.) in its deployed TPD (tamper-proof device), which is secured by the vehicle manufacturer [5]. A secure communication medium between vehicles, RSUs and GTA is assumed. Note that as vehicle A_u communicates with different sources, vehicle tracking and positioning is hard to achieve.

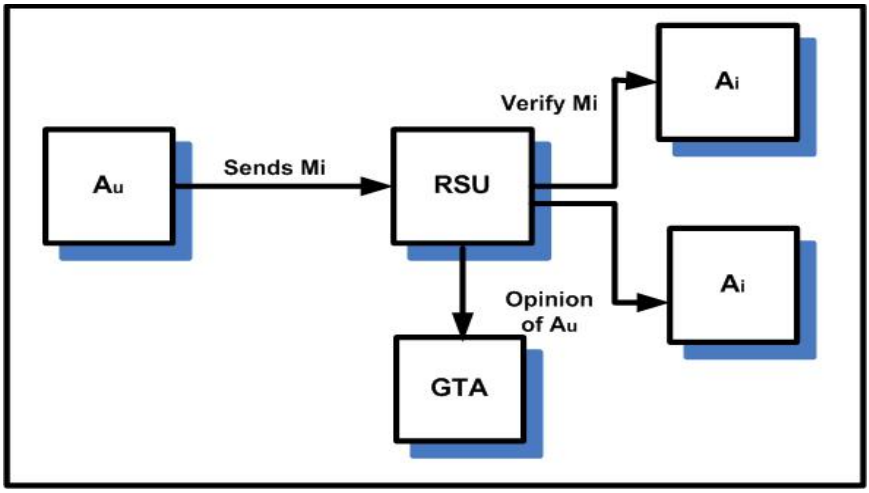


Fig. 1. Verifying M_i

Node categories for trust management purposes are presented in Table 1. The scheme divides user nodes into seven categories, each representing a specific percentage of trust range. The initial state of a new comer node, known as beginner node, is the C2 category. Contributing nodes in the system should be honest and active, in the sense of continuously reporting witnessed events.

Table 1. Node categories for trust management purposes

Node Category	Range of trust values / percentage	Users	Confidence Value
C1	$T_n < 0$	Deceptive / hackers	0
C2	$0 \leq T_n < 20$	Beginners / deceptive	0
C3	$20 \leq T_n < 40$	Weak/passive users	0.2
C4	$40 \leq T_n < 60$	Moderate users	0.4
C5	$60 \leq T_n < 80$	Active users	0.6
C6	$80 \leq T_n < 90$	Traffic reporters, ambulance, firefighting, etc	0.8
C7	$90 \leq T_n < 100$	Traffic authorities, police car, etc	0.9

Nodes categorized from C1 to C5, are the ordinary people that don't represent any official authorities, e.g. bus and taxi drivers work 24/7 on a daily basis, and covering nearly every street in different areas of the region. Thus, they could help in road monitoring, if they proved to be trustworthy. Therefore, if such nodes are active and honest in reporting events continuously, their trust values will increase and they will be promoted to reach C5 category. Such nodes will also be encouraged through giving incentives to them if they are in the C5 category. Such incentives could be discount in taxes, insurance charges, gas coupons, free parking vouchers, etc. Note, that maximum trust value a node could gain must be less than "0.8" or "80%". This is in order not to be categorized as an official authority. C6 and C7 nodes represent trusted authorities

which are pre-authenticated by GTA and thus there is no need to calculate their trust value. Accordingly, nodes which are categorized as C6 and C7 are the most trusted vehicles. In case an authorized node was compromised and was reported by RSU, its trust value will decrease or it might be blocked from accessing the network. The majority of traffic congestions and accidents take place at main streets in an area, with a high probability of the presence of one of the nodes between C5 to C7 categories. Thus, presence of C5 to C7 nodes addresses the scarcity problem since trusted nodes will be pre-defined.

In case an RSU is compromised, the nearest node that falls in these two categories (C6 or C7) will be the best to take over the job of RSU temporarily, until the problem is solved. This is known as delegation of authorities [14]. Node evaluation is done by RSU or nearby trusted authorities that fall in C7 category. C7 nodes can help in reporting events and other misbehaving nodes. If a C6 or C7 node got compromised and was reported by RSU or another C7 category, GTA have the right to degrade it or prevent it from logging into the network, so it can't further deceive other nodes.

3.3 Scheme Design and Trust Evaluation Steps

Let M_i be the message number. Node A_u sends message M_i to RSU or to any trusted authority, reporting an event. M_i must be first verified by RSU, before being broadcasted, this verification process primarily includes three checking steps:

$$\left.\begin{array}{l} \mathbf{0 \le L_{Max} \le 100} \longrightarrow \mathbf{(a)} \\ \mathbf{0 \le T_{Max} \le 20} \longrightarrow \mathbf{(b)} \\ \mathbf{0 \le M_{Max} \le 10} \longrightarrow \mathbf{(c)} \end{array}\right\} \quad (1)$$

First step is to check, L_{Max}; the location of the reporting vehicle. The distance between the reporting vehicle and the reported event should not exceed 100 meters in any direction, for efficiency and accuracy purposes. Second step is to check, T_{Max}; the time when a vehicle had sent a message reporting an event. The interval between the time the event took place and when reported, should not exceed 20 minutes for accuracy reasons. It could be of no use to report an old event, because it might have taken place in the past time and ended, since the topology of a VANET is very dynamic [2, 7]. M_{Max} is the maximum number of accepted messages by RSU concerning the same event which is the third checking step. This is in order to free the communication channel for other events to be reported and decrease the computation overhead and memory consumption at the RSU side.

Equation 1 highlights 3 conditions that have to be satisfied in order for an event to be identified, new event reported to RSU. The mentioned parameter could be adapted to different applications. The scheme moves to the second step of M_i evaluation. Let α_{Mi} be the weight of each message M_i. Weights for each message type are assigned by RSU during the verification process by opinion piggybacking process done by other nodes as shown in Table 2.

Table 2. Trust values for messages sent

M_i weight (α_{Mi})	Reason	M_i Situation
2	Honest witness	Approved/broadcasted
1	Honest report	Approved/broadcasted
0	Nonsence / meaningless / incomplete info / not confirmed	Discarded
-1	Dishonest report	Discarded
-2	Dishonest witness	Discarded

A sent message has a fixed standard format saved on all vehicles, that includes event location (estimated by GPS) and event time (system time), estimated time for event to end, road situation (fully blocked, partial blocked, not affected), involved vehicles if possible. If any these data is missing, the message is considered incomplete and thus is graded as zero. In addition, if the sent message was not approved by the RSU, it is graded as zero or "-1". In both cases, the message is discarded. Upon message evaluation by RSU, A_u credit (α_{Mi}) will be sent to GTA to be stored. Let M_t be the total weight of messages, which their trust value had not been yet processed as depicted in Equation 2,

$$\mathbf{M_t = \sum_{i=1}^{Mm} \alpha_{Mi} \quad where\ M_m \leq 50 \qquad M_t \leq 100} \tag{2}$$

Where M_m is the total number of unevaluated sent messages of node A_u, known as message counter. Note that M_m, should not exceed 50, or M_t should not exceed 100, because the function is divided by 100 to get it in percentage form. If any of these limits had been reached, a force trust update function will be activated, to compute the new trust value. In real life, it would not exceed this limit because trust values updates are done periodically during key renewal session. Initial values of M_t, M_i, and M_m, are zero.

In the previously introduced schemes [3], [4], new trust value of node A_u, $T_n(A_u)$ was calculated at every message received at the node level. Too much calculations leads to high processing overhead and memory consumption. This is not applicable in dense areas, since it is also time consuming. In a highly dynamic vehicular environment, it is essential to make quick correct calculations and consequently make decisions. Thus, every group of "m" messages will be evaluated, e.g. every five messages. After computing M_t, let $W_C(A_u)$ be the current trust weight of node A_u,

$$\mathbf{W_C(A_u) = (\sum_{i=1}^{Mm} M_t) / 100} \tag{3}$$

In Equation 3, $W_C(A_u)$ will be representing the summation of the "*m*" weighted messages. Initial value of $W_C(A_u)$ is zero. Upon getting the current trust weight of node A_u, new trust value will be computed with respect to A_u's trust history by RSU. Let $T_o(A_u)$ be the old trust value of A_u, and $T_n(A_u)$ is the new trust value of A_u. The Initial value of $T_o(A_u)$ and $T_n(A_u)$ is zero for beginner nodes. If A_u's old trust value if $T_o(A_u) \geq 0$, then A_u's new trust value $T_n(A_u)$, will be calculated as shown in Equation 4:

$$\mathbf{T_n(A_u) = T_o(A_u) + W_C(A_u)\ [(1 - T_o(A_u))] \quad if\ T_o(A_u) \geq 0} \tag{4}$$

Otherwise, if A_u's old trust value if $T_o(A_u) < 0$, then A_u's new trust value $T_n(A_u)$, will be calculated as shown in Equation 5:

$$\mathbf{T_n(A_u) = T_o(A_u) + W_C(A_u)\,[(1 + T_o(A_u))] \quad if\ T_o(A_u) < 0} \qquad (5)$$
$$\mathbf{where \quad -0.1 \leq T_n(A_u) < 0.8}$$

3.4 Advisory Network

The formation of an advisory network, is quiet a challenging objective. There could be available trusted nodes in a situation, but no one is able to reach them and distinguish them, because they are not declared trusted. For instance, if an RSU received a message claiming the blockage of a road due to an accident, the RSU needs to verify the event reported in message, and if true, it will be broadcasted. In addition, after the specified estimated time for event to end, RSU needs to get updated with the recent situation. Therefore, RSU should be able to search for the nearest highest available category of nodes, which could be C6 or C7 categories to give their opinion in this regards. If there weren't available nodes in these two categories, then RSU will have to search for other nodes in different categories from C2 to C5 category, giving higher priority to higher categories in condition they are nearer in location and time of event taking place, and ask them to give their opinion in regards to the mentioned event. This is called opinion piggybacking. Nodes are more likely to give their opinion on an event, because they know that they gain credit and privileges for so. Nodes will look forward to improve their trust values consequently and to receive advices from the network in situations when they are in need. Therefore, the development of an advisory network is an essential factor.

In this context, each node is assigned a confidence value, known as confidence building measure, which reflects its category and trust value as shown in Table 1. The confidence value reflects the percentage of trust in the sent information by a vehicle and is computed by RSU during message verification process. It is concatenated to node ID no. and sent with every message. This is to make it easy for recipient nodes to distinguish messages they receive easily and in real time. The confidence value is assigned by the GTA only during trust computations and is unchangeable by any other entity.

In case a C1 or C2 node sent its opinion about an event, since they are considered as independent nodes, they are given zero confidence values, in order to prevent them from deceiving other vehicles. Notice that the confidence value is a discrete number that represents node's minimum trust value of the category they are placed in. The node confidence value increases as its trust value and category is improved. Any announcement from C6, C7 or GTA is given the highest confidence value and doesn't need any further computation or opinion piggybacking to verify message content. While for nodes that sent a report claiming an event and lay in C1 to C5 categories, needs their confidence value to be computed by RSU to ensure efficient broadcasting, as follows:

Let $C_v(A_u)$ be the confidence value of node (Au), $C_{vt}(M_i)$ be the total confidence value of M_i,

$$\left.\begin{array}{l} C_{vt(yes)}(M_i) = (\sum_{i=1}^{m} Cv(Au)) \longrightarrow (a) \\ C_{vt(no)}(M_i) = (\sum_{i=1}^{m} Cv(Au)) \longrightarrow (b) \\ C_{vt}(M_i) = C_{vt(yes)}(M_i) - C_{vt(no)}(M_i) \longrightarrow (c) \end{array}\right\} \quad (6)$$
$$\text{where } 0 \leq C_v(A_u) \leq 1$$

Equation 6 is used to compute confidence values, in case there was not an authorized node (C6 or C7 nodes) available to witness an event. RSU computes the confidence value of gathered feedbacks of message M_i, for those nodes who claimed "Yes" and those nodes who claimed "No" for an event. It compares both results and takes the higher measure to be broadcasted. This is done for accuracy reasons and to eliminate the effect of deceptive replies sent and collusion attacks. Note that nodes in C1, C2, and C3 categories have a harder chance for their generated messages to be approved, in comparison to nodes in higher categories, like C4 and C5. Therefore, deceiver nodes could hardly reach their goals because they are given very low confidence measure.

C1 nodes are never consulted, since they are considered as deceptive nodes. For categories C4 to C7, they represent about 60% of the available nodes, thus RSU has a good chance to get correct information in a small amount of time, without having to calculate a majority opinion on reports sent. This forms the advisory network.

3.5 Penalty System

To monitor malicious nodes and keep track of their actions through time, a penalty system is presented. It consists of six different penalties. Each dedicated according to the severity of the action performed by the node. Penalties are P1, P2, P3, P4, P5 and P6.

If a node had violated any of the traffic rules and was reported by a C6, C7 node or RSU (e.g. breaking a traffic light, exceeding speed limit, wrong parking, etc.), RSU sends penalty P1_(A_u) to GTA to take action accordingly. In return, GTA decreases the trust value of node (A_u) by "0.02" (2%), as a penalty for its misbehavior as in Equation 7.

$$\textbf{P1:} \qquad T_n(A_u) = T_o(A_u) - 0.02 \quad (7)$$

If a node was proved to be accused in a crash accident, RSU sends penalty P2_(A_u) to GTA to take action accordingly. In return, GTA decreases its trust value by "0.1" (10%), as a penalty for its misbehavior, as in Equation 8.

$$\textbf{P2:} \qquad T_n(A_u) = T_o(A_u) - 0.1 \quad (8)$$

A strict monitoring for traffic violations and guilty accident nodes is being applied, by counting each of them per year. If they exceeded a specific limit, the protocol will be enforcing more penalties on node by decreasing its trust value $T_n(A_u)$ by "0.1" (10%) as highlighted in Equation 8.

Moreover, nodes that reached C5 category, have the right to benefit from the incentives placed. But C5 nodes could turn over to be lazy ones, as they have already

reached the maximum trust level they could gain. In contrary, they will be only receiving reports describing current road situation and not encouraged to help the system for a better performance. Therefore, in order to ensure that such nodes will still be active even though they reached the highest category. Let "n_n" be the number of driving days of node A_u that is counted by the vehicle's TPD. If $0.60 \leq T_n(A_u) < 0.80$ and M_m = "0" for $10 \leq n_n \leq 20$, then enforce penalty "P3" as in Equation 9.

$$\textbf{P3, P5:} \qquad T_n(A_u) = T_o(A_u) - 0.02 \tag{9}$$

Equation 7 was reused for penalty "P3", which is activated automatically by vehicle's TPD if the above conditions are satisfied. The initial value of "n_n" is zero and is only initialized by GTA during trust computation. Note that nodes who had not been driving for personal reasons will maintain their old trust values and category unchanged. Terminated nodes also maintain their last trust value unchanged. If a node attempt a late trust update for a certain number of driving days, penalty "P5" will be enforced. If "n_n" exceeded thirty to thirty-five, penalty P5 will be automatically activated by vehicle's TPD, that sends P5_(A_u) to GTA via nearest RSU, which by return decreases node's trust value and forces a trust update.

In case a node insists on its bad behavior and keeps on sending malicious reports and deceiving its neighbor nodes and RSU, its trust value will be decreased severely to reach C1 category. Assume that the minimum accepted trust value by the system is "-0.1", but if it keeps on this record, $T_n(A_u) \leq -0.1$ for $10 \leq n_n \leq 30$, then vehicle's TPD will report penalty P4_(A_u) to GTA via nearest RSU. GTA by return temporarily prevent node A_u from using the network as a penalty for certain number of days.

P4, P6: Temporarily blockage of node A_u from network

This mainly happens to nodes in C1 category which continuously claim wrong events. They are temporarily stopped from sending or receiving any road condition reports as penalty "P4" states. During message verification and trust computation, RSU can detect malicious nodes that don't satisfy system conditions. RSU sends to GTA complaining such nodes, which by return executes penalty P6_(A_u), to prevent it from using the network.

Isolating malicious nodes will increase the system efficiency. Since, it won't be fair to enjoy getting correct road situation reports and system benefits; meanwhile they harm others and irritate surrounding nodes with their false reports. Note that penalties P1, P2 and P6 are initiated by the RSU, whereas penalties P3, P4 and P5 are initiated by the vehicle's TPD.

If no answer is received M_i is saved until its estimated time ends and then discarded, $\alpha_{Mi}(A_u) = 0$.

- Nodes tend to report to RSU or C7 nodes, for privacy and confidentiality concerns of each node and to update their trust values.
- C6 and C7 nodes can help in broadcasting, in case near RSU was functionally overloaded.

- A consulting node choice is done with respect of category state giving priority to higher categories, then lower categories given that L_{Max} and T_{max} are satisfied.
- The maximum number of accepted messages "M_{Max}" in regards to the same event is ten. This is to prevent network congestion and decrease message verification computation overhead.
- To decrease message transmission overhead, GTA could compute M_t for all involved nodes, but keep it stored in its database for every node, without sending back to RSU and involved nodes.
- If RSU_1 received a report claiming a certain event in location which is within coverage range of RSU_2, then RSU_1 forwards M_i concatenated to it, M_i sender ID no. to RSU_2. Which in return starts M_i verification process as described previously.

3.6 Discussion

Deployed cameras on roads will help in traffic surveillance and message evaluation. For instance, upon the claim of an event, RSU can send to nearest deployed camera and checks for available relative snapshots. If the event being investigated is within its range, its feedback would increase evidence evaluation accuracy.

In case a node had a good trust history and is categorized as C4 or C5 node. If for some reason, this node in a certain situation had a personal benefit to collude with other nodes, which also have a good trust history, to achieve a personal goal. Even if one honest node claimed the truth, against seven or eight dishonest ones, these nodes together will gain a higher confidence value for their claimed event, due to their good history. But if they were discovered by a trusted authority or RSU surveillance process, they could be severely penalized. Though this situation will not occur very frequently, it is important to realize it.

In another case, a small percentage of drivers would acquire an international driving license beside their local one, e.g. truck drivers travelling from one country to another. In such a case, if both licenses were of no relation to each other, a misbehaving driver would take advantage, and drive with the more reputable driving license. Therefore, it is recommended that all the local license driving data should be completely linked to the international license. In other words, both licenses' data should be concatenated together, where any information updating or changes in one of them, should be automatically reflected in the other license. In this way, both driving licenses will work as one. Since it is the same user, the behavior tendencies will never differ. This also makes fewer burdens on GTA and RSU, for generating and initializing new variables for a new user.

3.7 Advantages and Comparison versus Related Work

The proposed scheme captures a history record of each node in the network in a central database for future processing. This history record is computed in a secure and accurate environment that allows the protocol to distinguish between different types

of nodes (authorized, honest, malicious, etc.). Trust computation is not limited to time or place, in fact, it is done dynamically and in a decentralized manner. RSU and GTA are the responsible entities that deal with node's data in trust computing, to guarantee node's privacy.

The scheme's penalty system consists of six different penalties, in an attempt to monitor all types of misbehaving actions committed by nodes covering different aspects in the vehicular network. Each node is punished according to the severity of the action it was accused for. The protocol keeps track of the nodes' number of accidents and traffic violations, and degrades its trust value, confidence value and category level accordingly. Therefore, it provides a real time trust computation, which is automatically updated. Effective transportation statistics could be extracted from such database.

No malicious node could claim to be a new comer node or another node, because the protocol works on the driver ID no., which is unique to everyone. Each broadcasted message also contains vehicle's ID no., which is only revealed by authorized entities. In previous trust computation schemes [3, 4], malicious nodes could take advantage that their "bad" history is not being monitored while they enter a new area or province and start repeating their bad behavior, such as sending untrue reports, break traffic lights, etc. In the categorized protocol, each node is placed in a certain category and has a confidence value that reflects its behavior through its driving life time. It is an automated system that guarantees fairness and accuracy, considering nodes privacy and confidentiality. Its category and confidence value are concatenated to its driving license ID no. throughout its lifetime and wherever it travels at any time. This provides easy and early malicious node detection and protects reputable nodes from being deceived.

This efficient message verification procedure done through computing the confidence values in Equation 6, guarantees high accuracy in broadcasting an event to nodes in the network, by computing the confidence value associated with each message. This also reflects the degree of trustworthiness of this node in the system. Each message should pass the stated conditions, before being broadcasted to other nodes in the network. Misleading reports are discarded to save memory and prevent them from future access by any compromised entity.

The scheme makes good use of honest nodes in the purpose of establishing an advisory network. Nodes are encouraged to participate in the network by giving their opinion on claimed events and gain credit for it, in the form of incentives for their honest responds. This solves the data scarcity problem, while establishing a scalable advisory network. Number of feedbacks upon a request, is limited to minimize computation overhead and communication channel jamming. The protocol prohibits malicious nodes from using the network, if they maintain their "bad" behavior. Consequently, nodes' confidence in the vehicular network environment is increased.

The scheme provides real time access, identification of malicious nodes; while for advisory purposes a node could reach authorized entities in real time. Regular vehicle's TPD and RSU tamper checks are done by using credentials and digital signatures to optimize system security. Thus, GTA checks that RSU is not compromised through its certificates and digital signature, before sending it node's data for trust computation. Comparison with previous schemes is stated in Table 3. The comparison table is inspired by the work in [7].

Table 3. Comparison versus related work

	Raya et al.[17]	Dotzer et al. [20]	Golle et al. [19]	Minhas et al. [3]	Chen et al. [10]	Gerlach et al. [9]	Patwardhan et al. [18]	Proposed CTMR
Decentralized	√	√	√	√	√		√	√
Scarcity			√	√	√	√	√	√
Dynamics	√	√		√	√	√	√	√
Scalability				√	√			√
Confidence	√			√	√	√		√
Security	√		√	√	√	√	√	√
Privacy		√	√	√		√		√
Robustness			√					√
History-Sensitive								√

4 CTMR Validation

Analysis and validation was performed using Matlab 7.10-2010a. Two case studies are shown in sections 4.1 and 4.2 to evaluate nodes' trust, malicious node detection and advisory nodes. In case study 1, assume nodes $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ are sending reports in regards to the events they witness. Nodes $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ will be evaluated every 5 messages they send "M_m=5", in order to save time and processing overhead as depicted in Table 4.

Table 4. Case study 1

Node	$T_o(A_u)$	Values of 5 messages (αMi)					n_n – days	W_C	$T_n(A_u)$	$C_v(A_u)$
		M_1	M_2	M_3	M_4	M_5				
A1	0.2	1	1	-2	1	1	5	0.02	0.216	0.2
A2	0.21	1	-1	0	-1	1	6	0	0.21	0.2
A3	0.38	1	1	1	0	2	2	0.05	0.411	0.4
A4	0.59	1	0	1	1	0	5	0.03	0.6023	0.6
A5	0.21	-1	-2	-1	1	-1	10	-0.04	0.1784	0
A6	0.79	-	-	-	-	-	10	0	0.77	0.6

4.1 Case Study 1

Summation of messages values will be calculated using Equation 2 (see section 3.3). The current trust weight of each node (W_C) will be calculated using Equation 3 (see section 3.3). Nodes $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ initial states $T_o(A_u)$, when this calculation took place are all above zero. Thus, new trust values $T_n(A_u)$ will be evaluated using Equation 4 (see section 3.3).

Note that node A_3, had old trust value "0.38", and in category C3, but after its good behavior, its new trust value reached "0.411", thus, it was promoted to C4 category. The same goes for node A_4 which was promoted from C4 to C5 category. Such fact is reflected in the confidence measure as well.

Meanwhile, node A_5 old trust value $T_o(A_5)$="0.21", but due to its wrong reporting that was monitored, its new trust value $T_n(A_5)$="0.1784". Note, that when $T_o(A_5)$ was "0.21", it was in category C3, but due to its incorrect reporting, it was degraded to category C2. Node A_6, is in C5 category, it became lazy were it didn't send any messages during the n_n-days (12), so it was penalized using "Equation 7", were its trust value was decreased by "0.02". Results for the second group of messages evaluated for nodes {A_1, A_2, A_3, A_4, A_5, A_6}, are shown in "Table 5". Nodes A_4, A_6 and could be consulted to give their opinion about an event. In contrast to nodes A_1, A_2, and A_5, because of their weak trust levels, advisory nodes are A_6, A_4, and A_3, respectively.

Table 5. Case study 1 (second group of messages)

Node	$T_o(A_u)$	Values of 5 messages (α_{Mi})					n_n – days	W_C	$T_n(A_u)$	$C_v(A_u)$
		M_6	M_7	M_8	M_9	M_{10}				
A_1	0.216	1	1	1	0	1	4	0.04	0.247	0.2
A_2	0.21	0	1	1	2	1	5	0.05	0.2495	0.2
A_3	0.411	1	2	1	1	0	3	0.05	0.44	0.4
A_4	0.6023	1	1	1	0	1	7	0.04	0.618	0.6
A_5	0.1784	1	0	-2	-1	-1	8	-0.03	0.1538	0
A_6	0.77	-	-	-	-	-	12	0	0.75	0.6

4.2 Case Study 2: Injecting New Comer and Malicious Nodes

The following scenario is assumed where nodes {A_7, A_8, A_9, A_{12}} are malicious nodes that fall in C1 category, and nodes {A_{10}, A_{11}} are beginner nodes, as shown in Table 6.

Table 6. Case study 2

Node	$T_o(A_u)$	Values of 5 messages (α_{Mi})					n_n - days	W_C	$T_n(A_u)$
		M_1	M_2	M_3	M_4	M_5			
A_7	-0.05	1	1	-2	1	1	4	0.02	-0.031
A_8	-0.1	-2	0	-1	-1	1	3	-0.03	-0.0127
A_9	-0.01	-1	-1	0	1	-1	5	-0.02	-0.03
A_{10}	0	0	-1	0	-1	1	3	-0.01	-0.01
A_{11}	0	0	0	1	1	1	6	0.03	0.03
A_{12}	-0.1	0	0	-1	-2	-1	10	-0.04	-0.1384

Nodes A_7, A_8, A_9 and A_{12}, have initial trust values below zero. Therefore, to evaluate their new trust value, Equation 5 was used. Whereas, A_{10} and A_{11} are beginner nodes with initial trust values zero, so "Equation 4 was be used, to calculate their new trust values after their interactions. Results for the second group of messages evaluated for these nodes are shown in Table 7.

Table 7. Case study 2 (second group of messages)

Node	$T_o(A_u)$	Values of 5 messages (α_{Mi})					n_n –days	W_C	$T_n(A_u)$
		M_6	M_7	M_8	M_9	M_{10}			
A_7	-0.031	-1	0	1	-1	-1	4	-0.02	-0.0506
A_8	-0.0127	0	-1	0	-1	0	10	-0.02	-0.0323
A_9	-0.03	1	0	-1	1	0	5	0.01	-0.0197
A_{10}	-0.01	-1	-1	0	1	-1	4	-0.02	-0.0296
A_{11}	0.03	1	0	0	1	1	6	0.03	0.0609
A_{12}	-0.1384	0	1	-2	0	-1	12	-0.02	-0.1556

All nodes except node A_{11} fall in C1 category, because of their malicious actions on the network. Those nodes could face a temporarily blockage as a penalty, if they continue in their bad behavior and exceed their limit of n_n driving days. Their confidence value is zero. If any of these nodes sent a report claiming an event, this report has a harder chance to be approved in comparison to trusted nodes of higher categories. Unless an authorized entity approves the report content to be true and valid, the message will be graded "0" and discarded after the end of its estimated time. Even if all those malicious nodes collude together in an aim to report an incorrect event, all their generated reports will not pass the confidence measure condition in Equation 6. This solves the collusion problem. It also guarantees correct information to be broadcasted, gives higher dependence on the system and its announcements, secures the network users against malicious attackers and selfish nodes.

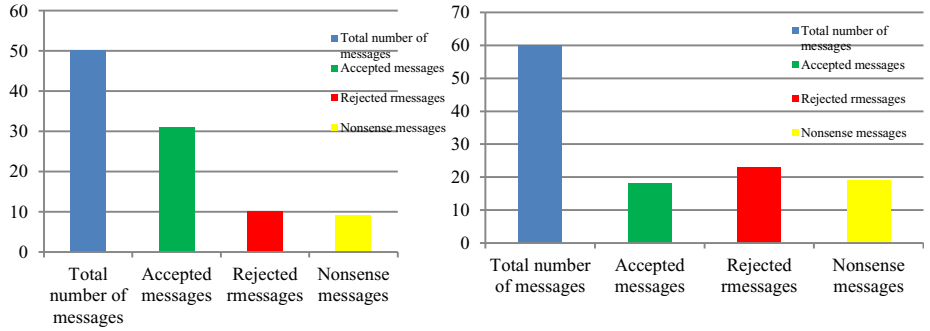


Fig. 2 (a). shows messages sent in Case study 1 and **Fig. 2(b).** Messages sent in Case study 2

Fig. 2(a) and Fig. 2(b) show the messages sent in case study 1 and 2, respectively. In case study 1, total number of sent messages is 50, 31 messages were accepted, 10 were rejected and 9 were nonsense messages. Malicious and nonsense messages are discarded in order not to be broadcasted to other nodes and to save memory space. Whereas, messages sent in case study 2 are shown in Fig. 2(b).

5 Conclusion and Future Work

Avoiding traffic congestion on roads is the key objective of vehicular networks. Security and trust are the key challenges in vehicular networks. Many researches were performed in trust management, in an effort to optimize network reliability and driving safety.

This paper presents a new scheme of categorized trusted communication scheme which is decentralized and can overcome attacks mentioned in section 2.1, and maintain message reliability. The protocol keeps a history record of each node in the network accompanied with penalties imposed on node. Nodes know that their interactions are being monitored, and a history record of their trust values is reflected in their category level and confidence value (as shown in Tables 3, 4, 5 and 6), which are concatenated to the driver ID license number. A malicious node will have a low trust category and confidence value, where its generated report could be discarded if it did not satisfy the proposed criteria, with a very low or no impact on the network. This guarantees early malicious nodes detection. The categories protocol solved the problem of coalition, where a group of familiar nodes report a wrong event to RSU in the aim of freeing a path for themselves, etc. Each node will be punished for its deceiving sent message, by being degraded from its high trust category to a lower trust category through the penalty system. Simulation showed accurate results for each node trust evaluation. Chain of trust is built within components of the network. The protocol considers only opinions from trusted nodes by the advisory network and discarding reports generated from distrusted nodes by the message verification process. It indicates the extent of nodes' trustworthiness. In addition, it guarantees to recipient nodes that messages they received are reliable and trustworthy. It also prevents network congestion by disabling malicious nodes from logging into the network. More exploration of the categorized system in ways to benefit from it in more real life applications is left to future work. Expanding the system in the basis of knowing nodes history and emphasizing on dedicating tasks to honest and authorized nodes in order to keep RSU and GTA resources for more critical and confidential applications, e.g. delegation of authorities, choosing an appropriate group leader, etc., are left for future research. Further performance evaluation including investigating the scalability of CTMR in terms of space and time complexity is the target of future work.

References

1. Qian, Y., Lu, K., Moayeri, N.: A Secure VANET MAC Protocol for DSRC Applications. In: Gobal Telecommunications Conference, LA, USA, pp. 1–5. IEEE (2008)
2. Huang, Z., Ruj, S., Cavenaghi, M., Nayak, N.: Limitations of Trust Management Schemes in VANET and Countermeasures. In: International Symposium on Personal, Indoor and Mobile Radio Communications, Toronto, Canada, pp. 1228–1232. IEEE (2011)
3. Minhas, U.F., Zhang, J., Tran, T., Cohen, R.: Towards Expanded Trust Management for Agents in Vehicular Ad-hoc Networks. International Journal of Computational Intelligence: Theory and Practice (IJCITP) 5(1) (2010)

4. Minhas, U.F., Zhang, J., Tran, T., Cohen, R.: Intelligent Agents in Mobile Vehicuar Ad-hoc Networks: Leveraging Trust Modeling Based on Direct Experience with Incentives for Honesty. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Toronto, Canada, vol. 2, pp. 243–247 (2010)
5. Rahman, S.U., Hengartner, U.: Secure Crash Reporting in Vehicular Ad hoc Networks. In: Proceedings of 3rd International Conference on Security and Privacy in Communications Networks, pp. 443–452. IEEE Computer Society (2007)
6. Ma, S., Wolfson, O., Lin, J.: A Survey on Trust Management for Intelligent Transportation System. In: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science, IWCTS 2011, pp. 18–23 (2011)
7. Zhang, J.: Trust Management for Vanets: Challenges,Desired Properties and Future Directions. International Journal of Distributed Systems and Technologies, 48–62 (2011)
8. Sumra, I.A., Hasbullah, H., Lail, J., Rehman, M.: Trust and Trusted Computing in Vanet. Computer Science Journal 1(1) (2011)
9. Gerlach, M.: Trust for Vehicular Applications. In: International Symposium on Autonomous Decentralized Systems Proceedings, AZ, USA, pp. 295–304 (2007)
10. Chen, C., Zhang, J., Cohen, R., Ho, P.: A Trust-based Message Propagation and Evaluation Framwork in VANETs. In: Proceedings of the 2nd International Conference on Information Technology Convergence and Services (ITCS), Cebu, Philippines, pp. 62–69. IEEE (2010)
11. Chuang, M., Lee, J.: TEAM:Trust Extended Authentication Mechanism for Vehicular Ad-hoc Netowrks. In: Consumer Electronics, IEEE International Conference on Communications and Networks (CECNet), Yichang, China, pp. 1758–1761 (2011)
12. Sumra, I.A., Hasbullah, H., Ahmad, I., Manan, J.B.: Forming Vehicular Web of Trust in VANET. In: Electronics, Communications and Photonics Conference (SIECPC), Riyadh, Saudi Arabia, pp. 1–6. IEEE (2011)
13. Sumra, I.A., Hasbullah, H., Ahmad, I., Manan, J.B.: New Card Based Scheme to Ensure Security and Trust in Vehicular Communications. In: Electronics, Communications and Photonics Conference (SIECPC), Riyadh, Saudi Arabia, pp. 1–6. IEEE (2011)
14. Biswas, S., Misic, J., Misic, V.: ID-based safety message authentication for security and trust in vehicular networks. In: Proceeding in International Conference on Distributed Computing Systems Workshops, Minnesota, USA, pp. 323–331. IEEE (2011)
15. Zhang, J.: A Survey on Trust Management for VANETs. In: International Conference on Advanced Information Networking and Applications, Biopolis, Singapore, pp. 105–112. IEEE (2011)
16. Abumansoor, O., Boukerche, A.: Towards a Secure Model for Vehicular Ad-hoc Networks Services. In: Global Telecommunications Conference (GLOBECOM), pp. 1–5. IEEE (2011)
17. Raya, M., Papadimitratos, P., Gligor, V.D., Hubaux, J.: On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks. In: The 27th Conference in Computer Communications, INFOCOM, USA, pp. 1238–1246. IEEE (2008)
18. Patwardhan, A., Joshi, A., Finin, T., Yesha, Y.: A Data Intensive Reputation Management Scheme for Vehicular Ad hoc Networks. In: International Conference on Mobile and Ubiquitos Systems-Workshop, California, USA, pp. 1–8 (2006)
19. Golle, P., Greene, D., Staddon, J.: Detecting and Correcting Malicious Data in VANETs. In: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, Philadelphia, PA, USA, pp. 29–37 (2004)
20. Dotzer, F., Fischer, L., Magiera, P.: A Vehicle Ad-hoc Network Reputation System. In: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Giardini Naxos, Italy, pp. 454–456 (2005)

Blind Watermark Approach for Map Authentication Using Support Vector Machine

Mourad Raafat Mouhamed[1,6], Hossam M. Zawbaa[2,6], Eiman Tamah Al-Shammari[3], Aboul Ella Hassanien[4,6], and Vaclav Snasel[5]

[1] Helwan University, Faculty of Science, Cairo, Egypt
mouradraafat@yahoo.com
[2] BeniSuef University, Faculty of Computers and Information, BeniSuef, Egypt
hossam.zawbaa@gmail.com
[3] Faculty of Computing Science and Engineering, Kuwait University
dr.eiman@ku.edu.kw
[4] Faculty of Computers and Information, Cairo University, Egypt
[5] VSB-Technical University of Ostrava, Czech Republic
[6] Scientific Research Group in Egypt (SRGE)
http://www.egyptscience.net

Abstract. This paper presents a blind and robust watermark approach for authentication 2D Map based on polar coordinates mapping and support vector machine is presented. The proposed system is composed of three phases. Firstly, in the preprocessing phase, the proposed algorithm mapped all vertices into polar coordinate system. Then, in the support vector machine phase, the watermark portable points will be chosen using support vector machine to reduce the number of these points which increases the imperceptibility without any effect on the robustness of the watermark. Afterwards, in the watermarking algorithm phase, the watermark is embedded into the map vertices using the random table of the decimal valued of the polar coordinates through the digit substitution of the decimal part. Finally, in the theoretical analysis and experimental results shows that the presented approach is robust against a various attacks including rotation, scaling, and translation. The proposed approach attained high imperceptibility.

Keywords: Support vector machine, vector watermarking, authentication, and geographic information system.

1 Introduction

2D Vector data has a very useful application like computer aided design (CAD) and geographic information system (GIS) where it cost a huge amount of money and time to collect data as in GIS or to design as in CAD. vector maps based on GIS data are rigorous representations of a geographical region and are used for many purposes such as military and civil cartography, urban planning, forestry etc. In those maps, each geographical structure, as a roads , a river or mountain, is defined by a definite number of vertices set in a specific arrangement.

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 84–97, 2013.

Production of GIS data is a heavy work. Thus, GIS data constitute a valuable asset which should be protected from digital piracy. The rapid growth of digital technology makes the modification, illegal copy and attack the digital data is simply moreover of intense image processing tools have also made digital image manipulations much easier. In such application of 2D vector data the demands of integrity and authenticity are very tough, and no deformation is permitted [1]. Recently, research on watermarking is concentrated on raster images [2], [3], [4], [5], [6], and [7].

When watermarking GIS data, we have to keep the data distortion low, i.e. the value of the coordinates of the vertices which define Map entities must to be very closely to the value of the coordinates in original map. There are two watermarking techniques when the researcher deal with vector data first one is transform domain techniques and the other is the spatial domain. the transform domain techniques first translate the spatial data into a transform domain and then apply the watermark in that domain The spatial domain techniques deal directly with the coordinates of vertices [9] [10]. Various transforms, like the Fourier descriptors [11] [12], wavelet transform [13] or mesh-spectral domain [14] have been used in the literature.

The watermarking of vector graphics have been developed in and several researches such as changing line features, insertion new vertices, and replacing existing stroke segments by new lines in a stylistic way are described in detail. They can achieve high capacity and robustness, but the watermark can be easily removed by attacks designed specifically for each method. A method for hiding data in curves has been proposed in [15]. It parameterizes a curve using the B-spline model and adds a spread spectrum sequence to the coordinates of the B-spline control points. It is robust against various attacks, such as collusion, cropping, geometric transformations, vector/raster (raster/vector) conversions, printing-and-scanning and some of their combinations. But it requires the original image for integrity verification, i.e., it is non-blind [16].

Jungyeop Kim in [17] used polygon vector feature to embed his watermark in interior angles where it has a weak point that he cannot extract watermarks if the interior angles are changed in our proposed method we use point vector feature to embed our watermark and it was robust for versus attack as translation,rotation and scaling. There is a big challenge to protect the vector map from illegal copyrights and from attacks that can destroy the benefit of the map by changing places and coordinates on it. This will lead to a great loss in confidential data and cost to be reconstructed. This paper presents a blind watermark approach that achieves the authentication in 2D vector map.

The rest of the paper is organized as follows. Section 2 gives an overview of the features of 2D spatial data. Section 3 describes the proposed watermarking map authentication approach including insert and extract watermark processes. Also, the calculation of mapping cartesian to polar coordinates and using the support vector machine. Section 4 discuss different attacks including rotation, translation and scaling. Section 5 presents the experimental result. Section 6 addresses conclusions and future work.

2 Features of Spatial Data: Background

Maps based on GIS data may be represented in spatial or transform domain in our work we will operate with spatial data where it be expressed as vector data. The vector data model represents each surface as a series of isolines; for example, elevation would be represented as a series of contours. However useful for displaying information, it does not easily support the calculation of surface characteristics such as the slope of the surface at a particular point, or the direction that the slope is facing. Both of these characteristics are important for analysis involving surfaces [9].

The road map is a real example of spatial data. A road map is a 2-D representation of object that contains points, polygons, and lines that can represent cities, roads, and political boundaries such as states or provinces. A road map is a imagination of geographic information. The location of cities, roads, and political boundaries that exist on the surface of the Earth are projected onto a 2-D display or piece of paper, preserving the relative positions and relative distances of the rendered objects [25].

The data that indicates the earth location such as (latitude and longitude, height and depth) of these rendered objects is the spatial data. For the rendered map, spatial data is used to project the locations of the objects on a two-dimensional. A GIS is often used to store, retrieve, and render this earth-relative spatial data. The feature of spatial data has its entity representative of using geometry which is built of one or more connected vertices, for more details refer to [8].

3 The Proposed Blind Watermark Approach for Map Authentication

The blind 2D vector watermark approach composed of three fundamental phases as follows:

- *Preprocessing:* The original map data is read as cartesian coordinates (x, y) and it will be converted into polar coordinates (r, Θ). Then, the data will divide into two groups one of them will be the training set and the other will be the testing set.
- *Support vector machine:* classifying data and define a little number of points to add the watermark. Support vector machine (SVM) success to find all watermark points at the extracting process.
- *Watermarking algorithm:* the watermark will be embed within the map image and the proposed method is blind to verify the watermark existing.

Fig. 1 and Fig. 2 depicts the building phases of the proposed system. These phases are described in detail in this section along with the steps involved and the characteristics feature for each phase.

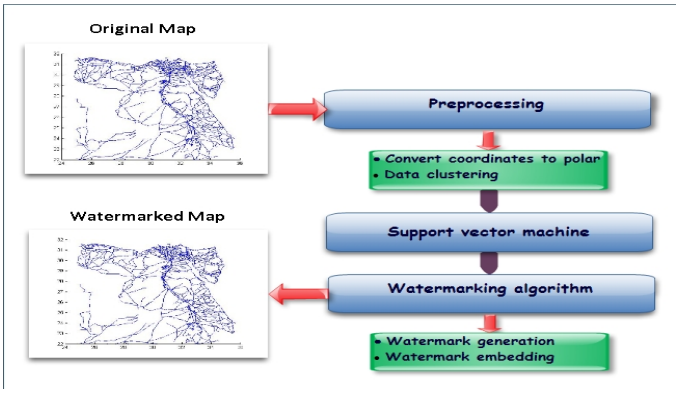


Fig. 1. The watermark embedding process

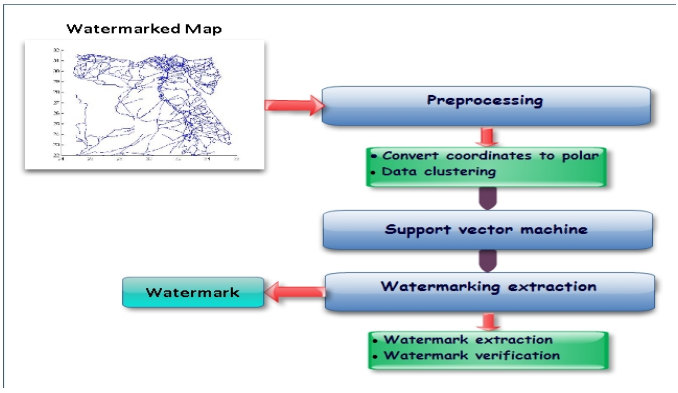


Fig. 2. The watermark extraction process

3.1 Preprocessing

Convert to Polar Coordinates. Mapping the cartesian coordinate system which has an origin point (x_0,y_0) then any point (x, y) in this system can be changed into polar coordinate system to be (r, θ) by the following equations 1 and 2.

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{1}$$

$$\Theta = \frac{y - y_0}{x - x_0} \tag{2}$$

In Figure 3, the origin of this system is $(0, 0)$ so that r expressed the distance between the point $(2, 3)$ and the origin and theta the angle between this radius

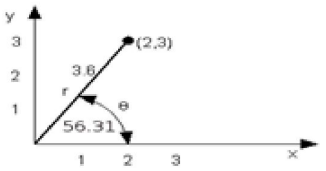


Fig. 3. Mapping from Cartesian to Polar Coordinates

and the x-axis, so that this point in polar coordinate expressed as $(3.6, 56.31)$. But to find this point in Cartesian coordinates again it has to use the following equations 3 and 4:

$$x = r * cos(\Theta) \tag{3}$$

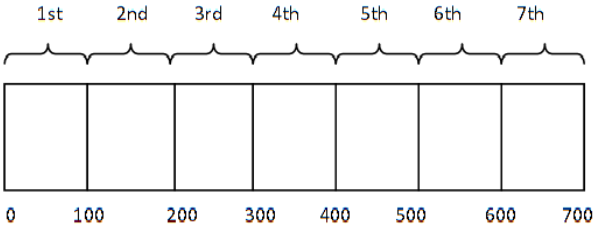
$$y = r * sin(\Theta) \tag{4}$$

Data Grouping and Clustering. The SVM algorithm need to group of data to deal with them and make the matching or classifying, so that the all data r and Θ will sorting descending or ascending separately then all of them will be divided into to group A and B. Group A will be divided to (n) intervals and the border of each interval will be the training data set (i.e the number of border in each point will be (n+1)). This process will happened with group B and the border of the intervals will act the test data set. Finally, the training set and testing set will be the input of the support vector machine.

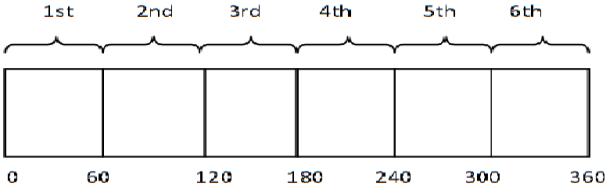
Fig. 4.(a) illustrate that if the the range radius (r) is [0, 700] the range will divided into seven intervals such as [0-100], [100, 200], .., [600-700]. The interval borders (0, 100, 200, 300, ... , 700) will be used to be the training data set or test data set. Fig. 4.(b) show the seme operation, but this operation for finding the borders of Theta (Θ) intervals.

3.2 Support Vector Machine

The efficiency of any watermark approaches measure by the imperceptibility and robustness. The imperceptibility measure by the random mean square error (RMSE) where it expressed the difference between the original and watermarked map and in the spatial data RMSE depend on the number of vertices that will be change and the value of this change. So, if we decrease the number of vertices, the RMSE will be decrease and then the imperceptibility will increase. Machine learning techniques like (Fuzzy c-Mean, K nearest classifier, support vector machine,.. etc) are help in classifying data and define a little number of points to add the watermark. But, when we work with Fuzzy c-Mean (FCM) and K-Mean machine learning techniques, we were found there is randomize and this will lead to instability of the robustness in watermark extraction. Support vector machine (SVM) success to find all watermark points at the extracting process.



(a) Radius interval borders



(b) Theta interval borders

Fig. 4. Interval Borders

SVM has emerged in recent years as a popular approach to the classification of data. SVM is margin-based classifier with good generalization capabilities [18]. It is the method of creating functions from a set of labeled training data. Given a training data set with n samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i is a feature vector in a v-dimensional feature space and with labels $y_i \in -1, 1$ belonging to either of two linearly separable classes C_1 and C_2. The function can be either a classification function or a general regression function. SVM finds an optimal separating hyper-plane between data points of different classes in a high dimensional space. SVM decoding models are based on the structural risk minimization (SRM) principle from statistical learning theory. In kernel model selection, mostly iterative search is applied in order to optimize the parameters within a specified range [20]. Geometrically, the SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires to solve the optimization problem, as shown in equations 5 and 6.

$$maximize \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j . K(x_i, x_j) \tag{5}$$

$$Subject - to : \sum_{i=1}^{n} \alpha_i y_i, 0 \leq \alpha_i \leq C \tag{6}$$

Where, α_i is the weight assigned to the training sample x_i. If $\alpha_i > 0$, x_i is called a support vector. C is a regulation parameter used to trade-off the training

accuracy and the model complexity so that a superior generalization capability can be achieved. K is a kernel function, which is used to measure the similarity between two samples. Different choices of kernel functions have been proposed and extensively used in the past and the most popular are the gaussian radial basis function (RBF), polynomial of a given degree, and multi-layer perceptron. These kernels are in general used, independently of the problem, for both discrete and continuous data. In the recent work the support vector machine is used to carry the watermark points.

3.3 Watermarking Algorithm: Generation of Watermark

The polar coordinates of each point are (r, Θ) have decimal and integer part, the integer part will be used to generate watermarking by using the random table and own-value as a following equation 7 [17]:

$$w = own\_value - \frac{key}{1000} \tag{7}$$

Where own_value will be produced by using ASCII code and key will generate by changing the integer part using random table 1.

Table 1. Real and changed values

Real value	0	1	2	3	4	5	6	7	8	9
Changed value	6	13	11	1	16	19	7	5	8	4

An example for converting own word (HKD) to get own_value by using ASCII code standard. H: 1001000(71), K: 1001011(74), and D: 1000100(67). Eventually, own_value becomes 212(71+74+67). This own_value becomes essential value when watermarks are embedded and extracted. Another example for finding the key, when the integer part of coordinate is 13 then the key will be 131 where the real value of 1 will be 13 and 3 will be 1 [17].

Embedding Watermark. The change in CAD and GIS data does not effect if it be after the sixth decimal digit so the watermark will be embed in the sixth digit or more. The proposed approach is described in the algorithm (1):

Watermarking Extraction. Where the proposed method is blind then it doesn't use the original data will follow the same steps of embedding process to verify the watermark existing. Algorithm (2) shows the details of the extracting watermarking.

Algorithm 1. The watermark embedding algorithm

1: Input: The original map vertices in Cartesian coordinates (x,y)
2: Input: change coordinates to the center point of map(x_0,y_0)
3: For each point cartesian coordinates (x,y) will be changed into polar (r, Θ)
4: sorting polar coordinates (r, Θ)
5: applying SVM on the polar coordinates r, Θ separately and find support vectors index for each coordinate
6: For each point every index of each coordinates we find cartesian coordinates(i.e watermark portable (x,y)) and then change into polar (r, Θ)
7: For each point polar coordinates (r, Θ) value will multiply by 10^s (strength of watermark $s >= 6$)
8: For each (r, Θ) decimal part will be changed by watermark and then marked polar ($\acute{r}$, $\acute{\Theta}$) coordinates will be produced.
9: For each ($\acute{r}$, $\acute{\Theta}$) will be changed into Cartesian another time to be ($\acute{x}$, $\acute{y}$);
10: Output: The marked map in Cartesian coordinates ($\acute{x}$, $\acute{y}$)

Algorithm 2. The watermark extracting algorithm

1: Input: The watermarked map map vertices in Cartesian coordinates ($\acute{x}$,$\acute{y}$)
2: Input: change coordinates to the center point of map(($\acute{x_0}$,$\acute{y_0}$))
3: For each point cartesian coordinates ($\acute{x}$,$\acute{y}$) will be changed into polar ($\acute{r}$,$\acute{\Theta}$)
4: sorting polar coordinates (r, Θ)
5: applying SVM on the polar coordinates r, Θ separately and find support vectors index for each coordinate
6: Input : The marked map vertices in Cartesian coordinates ($\acute{x}$,$\acute{y}$)
7: For each point Cartesian coordinates ($\acute{x}$,$\acute{y}$) will be changed into polar ($\acute{r}$,$\acute{\Theta}$)
8: For each point polar coordinates ($\acute{r}$,$\acute{\Theta}$) value will multiply by 10^s (strength of watermark $s >= 6$)
9: For each ($\acute{r}$,$\acute{\Theta}$) decimal part will verify that the watermark is existing.

4 Watermark Verification

4.1 Translation Attack Verification

Given a vertex in a map with Cartesian coordinates (x,y), a translation by Δx and Δy in the x and y axes, respectively, leads to new coordinates ($\acute{x}$, $\acute{y}$) = (x+ Δ x, y+ Δ y). The corresponding new polar coordinates ($\acute{r}$, $\acute{\Theta}$) become:

$$\acute{r} = \sqrt{((x + \Delta x) - (x_0 + \Delta x))^2 + ((y + \Delta y) - (y_0 + \Delta y))^2} = r \tag{8}$$

$$\acute{\Theta} = \arctan \frac{((y + \Delta y) - (y_0 + \Delta y))}{((x + \Delta x) - (x_0 + \Delta x))} = \Theta \tag{9}$$

We can get $\acute{r}$ = r and $\acute{\Theta} = \Theta$, that means the angular and radial coordinates did not change. Due to the fixation of these coordinates' values we will found no change in watermark.

4.2 Rotation Attack Verification

Given a vertex with coordinates (x, y) in the Cartesian coordinate system, after rotating it by an angle α , the new Cartesian coordinates $(\acute{x}, \acute{y})$ as in the following equations 10 and 11:

$$\acute{x} = x * \cos(\alpha) + y * \sin(\alpha) \tag{10}$$

$$\acute{y} = y * \cos(\alpha) - x * \sin(\alpha) \tag{11}$$

After applying a polar transformation to $(\acute{x}, \acute{y})$, the radial coordinate becomes as in the equation 12:

$$\acute{r} = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{12}$$

Where (x_0, y_0) is the map center. We can get $\acute{r} = r$, which means the radial coordinate, is not changed. So the proposed method is invariant to rotation.

4.3 Scaling Attack Verification

Any vertex with coordinates (x, y), after a scaling attack by a same factor of $\mathbf{S}$ Cartesian coordinates will be $(\acute{x}, \acute{y})$ as in equations 13 and 14.

$$\acute{x} = x * \mathbf{S} \tag{13}$$

$$\acute{y} = y * \mathbf{S} \tag{14}$$

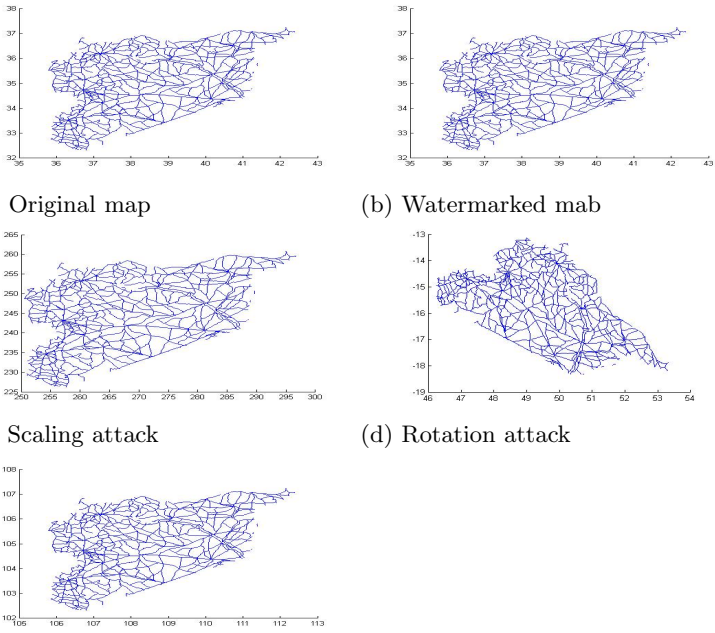
By changing into polar coordinates, we get the equation 15:

$$\acute{\Theta} = \arctan \frac{((y * \mathbf{S}) - (y_0 * \mathbf{S}))}{((x * \mathbf{S} - (x_0 * \mathbf{S}))} = \Theta. \tag{15}$$

Since $\acute{\Theta} = \Theta$, the angles before and after this attack are the same. As indicated in previous equation, it shows that the proposed method is invariant to scaling. Now it's proved that the proposed method is robust against this attacks.

5 Experimental Results

Our experimental result had been implemented with windows 7 professional (64 bit) and matlab 7.9.0 and the real data from DIVA-GIS site with different shape files. Now the original Syria and watermarked map showed at figure 5(a), 5(b), respectively followed by scaling, rotation, and translation attacked map in the figure 5(c), figure 5(d) and figure 5(e) respectively.



(a) Original map

(b) Watermarked mab

(c) Scaling attack

(d) Rotation attack

(e) Translation attack

Fig. 5. Syria map watermarking and attack

5.1 Analysis of Visibility

In the experiments, the parameters were chosen as follows: embedding strength $s = 6$, own_value is the corresponding ASCII standard code of word (Mourad), and the table 1. Here, the root mean square error (RMSE) as formulated in equations 16 and 17, are used to represent the distortion inflicted on the graphic by our watermarking approach.

$$RMSE = \frac{\sum \|V - V_w\|}{N - 1} \tag{16}$$

where

$$V = \sqrt{x^2 + y^2} \tag{17}$$

Where is N is the number of vertices, $\mathbf{V}$ and V_w are the original and marked vertex respectively. Tables 2, 3 expresses the relation between RMSE and the strength of watermarking of four maps Roma, Egypt,China and Syria without(i.e watermarked all vertices [23]) and with using SVM respectively and if we get Egypt roads as example we found that at strength s=8 when we didn't use SVM the RSME was 3.78E-09 from table (II) but when we improve this approach by using SVM we found that the RMSE was 1.48E-12 and that prove the superior of using SVM in the imperceptibility issue.

Table 2. Relation between watermark strength s and $RMSE$ without using SVM

Map	No. vertices	s=5	s=6	s=7	s=8	Execution time
Roma_roads	31986	4.024E-06	4.5358E-07	3.3736E-08	3.3922E-09	35.704692
Egypt_roads	30022	3.61E-06	3.57E-07	4.11E-08	3.78E-09	97.259982
China_roads	389442	6.67E-06	6.61E-07	6.64E-08	6.61E-09	906.244648
Syria_roads	11828	4.15E-06	3.27E-07	4.85E-08	4.61E-09	88.073596
Tunesia_roads	6533	3.9560e-006	3.9884e-007	4.0086e-008	4.0791e-009	15.977018
Turky_roads	70184	4.6485e-006	4.6450e-007	4.6771e-008	4.6767e-009	203.980169

Table 3. Relation between watermark strength s and $RMSE$ by using SVM

Map	No. vertices	s=5	s=6	s=7	s=8	Execution time
Roma_roads	31986	3.0158E-09	1.2012E-10	8.3363E-12	7.3786E-13	7.709183
Egypt_roads	30022	1.03E-09	1.44E-10	1.44E-11	1.48E-12	6.126416
China_roads	389442	6.27E-10	6.24E-11	7.47E-12	6.87E-13	461.154683
Syria_roads	11828	4.63E-09	3.91E-10	3.522E-11	3.70E-12	7.181789
Tunisia_roads	6533	1.6549e-009	3.5108e-010	1.6398e-011	1.6719e-012	7.093668
Turky_roads	70184	5.2360e-010	3.7549e-011	3.8138e-012	5.4038e-013	16.556669

The experimental result show there is a negative relation between the number of watermarked vertices and the imperceptibility measurement (RMSE), figure 6, Table 4 illustrate the relation between the number of vertices and the RMSE at watermark strength $s = 8$, ex: China-roads map when the number of watermarked vertices 26, 44, 58, and 75. The RMSE is $4.5695E-13$, $7.2842E-13$, $9.429E-13$ and $1.53E-12$ respectively that show that there is direct relation between the RMSE when the number of vertices increase that make an increasing in the RMSE. Also the experimental gave better execution time with SVM than without it for example, the time of embedding watermark in *China_roads* map at strength $s = 8$ without using SVM gave 906.244648 seconds, where when uses SVM it was 461.154683 and this verify that the using SVM in the technique reduce the execution time.

5.2 Robustness Analysis

The result of applying geometrical attacks (i.e translation, rotation, and scaling) on roads maps of Egypt, Syria, Roma and China, illustrate the power of robustness of this proposed approach and that proved by calculating the **NC** between the watermarked and original maps after the attacks. The normal correlation **NC** calculate by the following equation 18:

$$NC = \frac{\sum W \times \acute{W}}{\sqrt{\sum W^2 + \sum \acute{W}^2}} \tag{18}$$

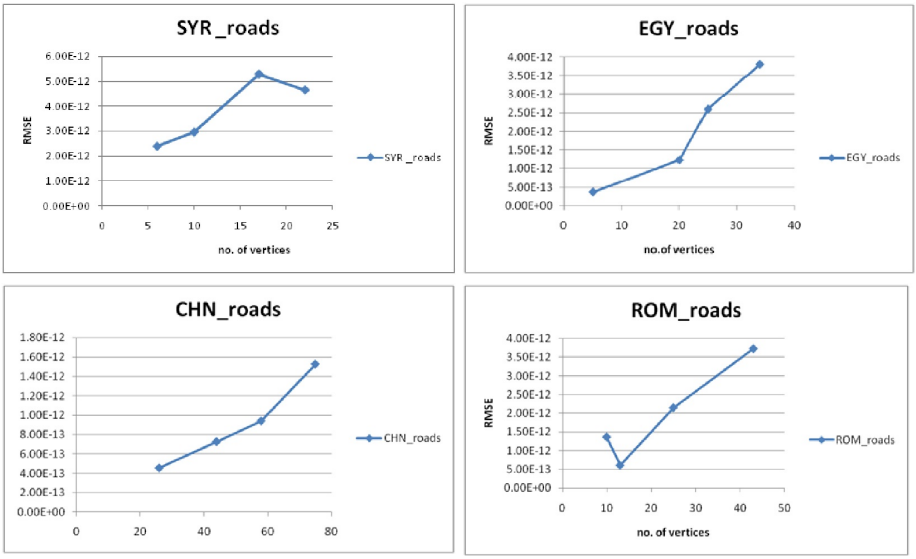


Fig. 6. Relation between Number of watermarked vertices and $RMSE$

Table 4. Relation between number of watermarked vertices and $RMSE$ without using SVM

	$no.of vertices$	$RMSE$
$Roma\_roads$	10	1.3644E-12
	13	6.0411E-13
	25	2.1503E-12
	43	3.7301E-12
$Egypt\_roads$	5	3.7064E-13
	20	1.2282E-12
	25	2.5982E-12
	34	3.7999E-12
$China\_roads$	26	4.5695E-13
	44	7.2842E-13
	58	9.429E-13
	75	1.53E-12
$Syria\_roads$	6	2.3896E-12
	10	2.9595E-12
	17	5.2718E-12
	22	4.6366E-12

Table 5. Relation between NC and versus attacks where S, R and T are Scaling, Rotation and Translation attack

Map	No.of vertices	S attack	R attack	T attack
$Roma_roads$	31986	1.00E+00	1.00E+00	1.00E+00
$Egypt\_roads$	30022	1.00E+00	1.00E+00	1.00E+00
$China\_roads$	389442	1.00E+00	1.00E+00	1.00E+00
$Syria\_roads$	11828	1.00E+00	1.00E+00	$1.00E+00$
$Turky\_roads$	70184	1.00E+00	1.00E+00	$1.00E+00$
$Tunisia\_roads$	6533	1.00E+00	1.00E+00	$1.00E+00$

Table 5 illustrates the power of robustness of the proposed approach with different maps with versus attack.

6 Conclusions

This paper presents blind and robust watermark approach for map authentication. It uses support vector machine and polar coordinates to embed watermark in its decimal value by using random table as a key. The experimental results show that the machine learning (SVM) help to increase the imperceptibility without change the robustness of watermark. this proposed method is robust against geometrical attack such equal scaling, rotation and translation and that can verify the authentication of this geospatial data where the good robustness guarantee that the watermark survive also after attack that can safe the ownership of data.

References

1. Shao, C., Wang, X., Xu, X., Niu, X.: Study on lossless data hiding algorithm for digital vector maps. Journal of Image and Graphics 12(2), 206–211 (2007)
2. Peng, F., Lei, Y., Long, M., Sun, X.: A reversible watermarking scheme for two-dimensional CAD engineering graphics based on improved difference expansion. Computer-Aided Design 43(8), 1018–1024 (2011)
3. Anusudha, K., Sangeetha, A.: Digital Watermarking of Satellite Images Using Secured Spread Spectrum Technique. International Journal of Recent Trends in Engineering 1(1) (May 2009)
4. Wolthusen, S.D.: Proceedings of the IEEE Workshop on Information Assurance (2006)
5. Celik, M., Sharma, G., Tekalp, M.: Lossless generalized-LSB data embedding. IEEE Transactions on Image Processing 14(2), 253–266 (2005)
6. Gaurav, B., Jonathan Wu, Q.M., Balasubramanian, R.: New robust adjustable logo watermarking scheme. Computers and Security 3(1), 40–58 (2012)

7. Nana, W., Chaoguang, M.: Reversible fragile watermarking for 2-D vector map authentication with localization. Computer-Aided Design 44, 320–330 (2012)
8. Suk-Hwan, L., Ki-Ryong, K.: A watermarking for 3D mesh using the patch CEGIs. Digital Signal Processing 17, 396–413 (2007)
9. Fei, P., Re-Si, G., Chang-Tsun, L., Min, L.: A semi-fragile watermarking algorithm for authenticating 2D CAD engineering graphics based on log-polar transformation. Computer-Aided Design 42, 1207–1216 (2010)
10. Kitamura, I., Kanai, S., Kishinami, T.: Copyright protection of vector map using digital watermarking method based on discrete Fourier transform. In: IEEE Geoscience and Remote Sensing Symposium (IGARSS 2001), vol. 3, pp. 1191–1193 (2001)
11. Kang, H., Kim Kab, I., Choi, J.: A map data watermarking using the generalized square mask. In: Proceedings of IEEE International Symposium on Industrial Electronics (ISIE 2001), vol 3, pp. 1956-1958 (2001)
12. Solachidis, N., Nikolaidis, I.P.: Fourier Descriptors Watermarking of Vector Graphics Images. In: Proceedings of IEEE International Conference on Image Processing, vol. 3, pp. 9–12 (2001)
13. Solachidis, V., Pitas, I.: Watermarking polygonal lines using Fourier descriptors. IEEE Computer Graphics and Applications 24(3), 44–51 (2004)
14. Ohbuchi, R., Ueda, H., Endoh, S.: Watermarking 2D vector maps in the mesh-spectral domain. In: Proceeding of Shape Modeling International, pp. 216–225 (2003)
15. Yuanyuan, L., Luping, X.: A blind watermarking of vector graphics images. In: Proceedings of Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003), pp. 424–429 (2003)
16. Zhang, W., Ren, X., Zhang, G.L.: An robust object locating method based on log polar transform and affine transform. Journal of Image and Graphics 11(9), 1255–1259 (2006)
17. Jungyeop, K.: Robust Vector Digital Watermarking Using Angles and a Random Table. Advances in Information Sciences and Service Sciences 2(4) (December 2010)
18. Syed, F.T., Asifullah, K.A., Anwar, M.M.: Support Vector Machine based Intelligent Watermark Decoding for Anticipated Attack. World Academy of Science, Engineering and Technology 21, 1001–1006 (2008)
19. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1-3), 131–159 (2002)
20. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University (2003)
21. Staelin, C.: Parameter selection for support vector machines. Technical report, HP Labs, Israel (2002)
22. Moghaddam, B., Yang, M.H.: Learning Gender with support faces. IEEE Transaction on Pattern Analysis and Machine Learning 24 (2002)
23. Mourad, R., Mouhamed, A.M., Hassanien, A.: Blind 2D Vector Data Watermarking Approach Using Random Table and Polar Coordinates. In: International Conference on Uncertainty Reasoning and Knowledge Engineering, pp. 67–70 (2012)
24. Wu, Q., Zhou, D.-X.: Analysis of support vector machine classification. Journal of Computer Analysis and Applications 8, 99–119 (2006)
25. Oracle® Spatial Developer's Guide, 11g Release 2 (11.2) E11830-11

High Payload Audio Watermarking Using Sparse Coding with Robustness to MP3 Compression

Mohamed Waleed Fakhr

Electrical and Electronics Department, University of Bahrain
Manama, Bahrain
mfakhr@uob.edu.bh

Abstract. A high payload audio watermarking technique is proposed based on the compressed sensing and sparse coding framework, with robustness to MP3 128kbps and 64kbps compression attacks. The binary watermark is a sparse vector with one non-zero element that takes a positive or negative sign based on the bit value to be encoded. A Gaussian random dictionary maps the sparse watermark to a random watermark embedding vector that is selected adaptively for each audio frame to maximize robustness to the MP3 attack. At the decoder, the Basis Pursuit Denoising algorithm (BPDN) is used to extract the embedded watermark sign. High payloads of (689, 1378 and 2756) bps are achieved with %BER of (0.3%, 0.5% and 1%) and (0.1%, 0.3% and 0.5%) for 64kbps and 128kbps MP3 compression attacks respectively. The signal to embedding noise ratio is kept in the range of 27-30 dB in all cases.

Keywords: Sparse Coding, Compressed Sensing, Audio Watermarking, MP3 Audio, Robust Watermarking, Basis Pursuit Denoising (BPDN).

1 Introduction

Robust watermarking techniques embed a secret code (watermark) within a multimedia file (e.g., MP3 music) where the watermark detection and verification is possible only for the certified authority. The watermark should be imperceptible, with minimal quality degradation (SNR in the vicinity of 30dB), should have high payload and robustness to attacks that do not destroy the original host signal, and should also be secure. This is a challenging problem for audio signals, and in particular, for MP3 audio where MP3 compression/decompression attack is inevitable. A large body of literature has emerged in the last few years dedicated to developing MP3 robust audio watermarking [1-8]. This is in part due to the challenges of the problem, and more importantly, due to the market need for a reliable copyright scheme to control the proliferation of MP3 music over the internet.

Robustness to MP3 attack, high payload and imperceptibility are conflicting requirements. Robustness requires more embedding strength, high payload requires more frequent embedding while imperceptibility requires high signal to embedding noise ratio SNR which is naturally lowered by the former two. Those conflicting criteria can be viewed as a multi-objective optimization problem, where one should

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 98–110, 2013.

ideally create an optimal watermark for each audio frame that would maintain the high SNR and be robust to the MP3 attack. And if the high payload is required, the frame length should be in the order of a few milliseconds. Reaching such optimal settings would require the watermark to adapt to the host signal characteristics and presumed attacks. This creates two problems: If the watermark is going to adapt to its host frame, how can this information be conveyed to the decoder? And if this adaptation mechanism is known and is not highly secure then the adversary can reproduce the watermark and eventually remove it.

One way to solve the security issue is to select the watermark embedding vector from a secret random codebook that exists at both the encoder and the decoder. However, the question remains at the detection stage for which watermark to look for within the codebook, since trying all of them at the decoder would lead to many errors. On the other hand, if the codebook is large enough, a search criterion at the encoder can be used to adaptively select the best watermark vector for each audio frame by employing an analysis-by-synthesis paradigm, thus pushing the complexity to the encoder rather than the decoder.

In this paper, a sparse coding watermarking technique that employs an overcomplete dictionary (codebook) is proposed by which the adaptive watermark can be detected seamlessly at the decoder while the security is maintained by the randomness of the codebook [9]. The watermarking technique proposed can be interpreted by the compressed sensing (CS) signal recovery paradigm or, by the sparse coding (SC) paradigm where the L1-minimization is used in both to estimate a sparse vector from its random projection.

At the encoder an algorithm searches for the best watermark vector "atom" in the codebook that is robust to the MP3 coding-decoding attack. Once the best codebook atom is identified, it is embedded in the host frame. At the decoder, the watermark detection is formulated as an L1-norm minimization basis pursuit denoising (BPDN) problem that is solved to estimate the sparse watermark. The rest of the paper is organized as follows. Section 2 briefs the related recent work in MP3 audio watermarking. In section 3, the compressed sensing and sparse coding related theory is discussed and the proposed watermarking idea is introduced. In section 4 the proposed adaptive watermark algorithm is introduced. Related work in CS watermarking is in section 5. Experimental results are detailed in section 6 and conclusions are drawn in section 7.

2 Audio Watermarking

MP3 is popular because it offers good audio quality with small storage. As a consequence, on-line stores have proliferated because the ease to exchange MP3 files in the Internet. Robust audio watermarking is one way of tracking and copyright management of audio files, where many different approaches have been proposed in the past few years [1-8]. Table 1 summarizes the best MP3-robust techniques among the recent ones for comparison with the results in this paper. Most of the proposed techniques use the quantization index modulation (QIM) and spread spectrum approaches

with variation on the embedding domain and exploiting audio characteristics. In summary, acceptable audio watermarking should have an SNR ≥30dB, a bit error rate (BER) ≤1% at MP3 attacks of 128kbps and 64kbp. The main challenge is to meet these specifications with the highest possible payload.

Table 1. MP3 Robust Watermarking Summary

Algorithm	MP3 Quality	SNR	%BER	Payl oad
Noriega [1] 2010	64kbps	40dB	0.013	230 bps
Bhat [2] 2010	64kbps	30dB	1	196 bps
Dhavale [3] 2011	32kbps	26dB	0.4	1378 bps
Yang [4] 2010	64kbps	30dB	0.02	22 bps
Hamdouni [5] 2012	64kbps	30dB	0.4	100 bps
Ercelebi [8] 2009	128kbps	30dB	0.5	170 bps

3 Proposed CS-SC Watermarking Framework

3.1 Compressed Sensing Related theory

Compressed sensing signal recovery relies on the concept of a sparse domain representation of compressible signals. In the basic formulation by Candès and others [10, 11], if a K-sparse vector x with dimension (N×1) is sampled with a random Gaussian M×N matrix Ω producing a measurement vector y with dimension (M×1) where M < N, then, given the measurement vector and knowing the sampling matrix Ω it is possible to recover the sparse vector x from y as follows:

$$y = \Omega x + e \tag{1}$$

Where e is a measurement noise with variance ε. An exact recovery of the sparse vector x is possible through L1-minimization using the basis pursuit denoising algorithm if Ω satisfies the restricted isometry property (RIP), which is met for Gaussian random orthonormalized matrices and many full of partial transforms [10, 11]. This is done by solving the convex linear optimization problem:

$$minimize\ \|\boldsymbol{x}\|_1\ s.t.\ \|\boldsymbol{y} - \Omega\boldsymbol{x}\|_2 \leq \varepsilon \tag{1}$$

Where ε is tolerance, and the following sparsity condition should be satisfied where K is the number of non-zero elements in x:

$$M > O\left(K\ Log\left(\frac{N}{K}\right)\right) \tag{3}$$

More recently, there has been an extension to this framework by Laska et. al. [12] in what was named the Justice Pursuit (JP) algorithm explained as follows: If the measurement vector is corrupted by an interference that is sparse in some domain, equation (1) becomes:

$$y = \Omega x + \Phi\beta + e \tag{4}$$

Where Φ is a full or partial transform or random matrix with orthonormal columns and dimensions (M×L) where $L \leq M$ and β is a sparse vector with possibly large amplitude non-zero components of length L and sparsity k. It was shown that both sparse vectors (x and β) can be recovered if the matrices Ω and Φ are incoherent and with uncorrelated columns, and the sparsity condition in (3) is now updated to:

$$M > O\left((K+k)\, Log\left(\frac{N+L}{K+k}\right)\right) \tag{5}$$

The recovery algorithm assumes a new sparse vector U=[x β] of size (N+L)×1 and a new matrix $\boldsymbol{\psi}$=[Ω Φ] with size M×(N+L), and (4) is re-written as:

$$\mathrm{y} = \psi\mathrm{U} + \mathrm{e} \tag{6}$$

And the basis pursuit denoising (BPDN) becomes:

$$minimize\ \|\mathbf{U}\|_1 \quad \text{s.t. } \|\mathbf{y} - \boldsymbol{\psi}\mathbf{U}\|_2 \leq \varepsilon \tag{7}$$

When the sparse vector U is estimated, its first N elements are those of x and the remaining L elements are those of β [12]. Many fast algorithms have been developed in the literature for solving the basis pursuit denoising, of those, the L1-magic library which uses the primal-dual algorithm [13] is used for all the L1-minimization linear programming algorithms in this paper.

3.2 Compressed Sensing (CS) Watermarking Paradigm

The above CS formulation lends itself directly to the watermarking problem. Let the watermark be a sparse vector β of length L and with k non-zero components which take the binary values ±1 based on the required watermark value. In this paper, k=1 (only one non-zero element). For embedding in a measurement vector of length M, a random vector watermark signal W is generated:

$$W = \Phi\,\beta \tag{8}$$

Where Φ is an orthonormalized iid random Gaussian matrix of size (M×L). The host audio signals (music or speech) are known to be compressible signals for which sparse or approximately sparse domains exist. Let the audio frame in the sparse transform domain be x which is assumed to be K-sparse (or approximately sparse) in some transform domain. Going back to the time-domain we take the inverse of the sparsifying transform, and the host signal in time domain is given by:

$$X = \Omega x \tag{9}$$

Where Ω is the inverse of the used transform matrix. Adding (8) and (9) we get the watermarked host signal given by:

$$X^w = \ y = \Omega x + \ \Phi\beta \tag{10}$$

Thus, given the watermarked signal X^w, the BPDN can be used to find both β and x.

3.3 Sparse Coding (SC) Interpretation

Sparse coding with adaptive and non-adaptive overcomplete dictionaries has been used in many applications such as image and audio denoising and classification [14-16]. The basic SC concept models a signal as a sparse decomposition of atoms from an overcomplete dictionary, where the atoms locations and strengths are estimated using the L1-norm minimization. Equation (10) may be generalized as follows:

$$\mathrm{X^w} = \ y \ = \Omega_1 x_1 + \ \Omega_2 x_2 + \cdots + \ \Omega_C x_C + \ \Phi\beta \tag{11}$$

The host signal in time domain is assumed to be a decomposition of C sparse vectors each in a different domain and the matrix Ω is given by $\Omega = [\Omega_1\ \Omega_2 \ldots \Omega_c \ldots \Omega_C]$, where Ω is now of size M×N, with N=C×M and each inverse transform matrix Ω_c is M×M. Equation (11) is written as:

$$X^w = \ y = \psi U \qquad where \qquad \psi = [\Omega\ \Phi] \tag{12}$$

Now ψ is the overcomplete dictionary of size M×(N+L), and U is the composite sparse vector its first N elements are those of x and the remaining L elements are those of β.

3.4 Watermark Embedding

Following (10), the watermarked host signal for the i_{th} frame is given by:

$$X_i^w = \Omega x_i + \alpha_i \ * \Phi\ \beta_i \tag{13}$$

Where α_i is an embedding strength which is adaptively adjusted to make the average SNR of the watermarked signal constant as follows*:* Since all watermark vectors (the columns of Φ) are normalized, the SNR is:

$$SNR = 10Log_{10}\ \frac{\sum_{i=1}^{M} Xi^2}{\alpha^2} \tag{14}$$

In this paper, the SNR is fixed at 30dB, and α is given by:

$$\alpha = 0.025\sqrt{\sum_{i=1}^{M} Xi^2} \tag{15}$$

The watermark sparse vector β_i contains only one non-zero element (the $j_{\mathbf{th}}$ element) with the required watermark sign. Thus, only one vector from the random matrix Φ is used: Φ_j for each audio frame. At the decoder, it is required to estimate the sign of the non-zero element in the β_i vector given the watermarked frame and the matrix Φ.

Having the measurement vector y, we apply the basis pursuit denoising algorithm as in (7) and the watermark sparse vector β is estimated as well as the sparse vector x and the estimated host signal in time domain is obtained by X= Ω x.

In the clean situation with no MP3 attack or other attacks, both the watermark location and sign are recovered perfectly with strong enough embedding strength (with SNR less than 70dB) and the number of columns in Φ is small. However, with the MP3 compression followed by decompression, more than one non-zero value usually appears in the recovered watermark and the sign of the correct location may be flipped. This is because the MP3 compression-decompression attack is highly nonlinear. Figure 1 illustrates a typical case where L=8 for the same audio frame and watermark, with and without the MP3 attack.

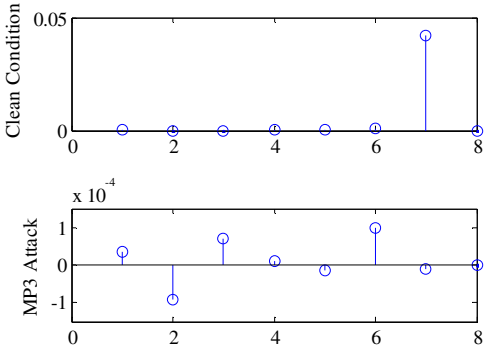


Fig. 1. Estimated sparse Watermark vector in MP3 attack

3.5 Watermark Extraction

Therefore, at the decoder, the largest non-zero value is found, and its sign is taken as the estimated watermark. This arrangement works well for small payloads and 128kbps MP3 quality. However, as shown in figure 1, it does not work well for 64kbps MP3 attack and for larger payloads as the correct spike may shift and more non-zero spikes appear. Moreover, it is very difficult to model the MP3 compression effect analytically on the recovered watermark due to the nonlinear and non-stationary behaviour of the MP3 attack.

4 Adaptive Watermark Selection Algorithm

Since there are L columns (atoms) in the Φ matrix it would be optimal to select the atom Φ_j for each new audio frame that is most resistant to the MP3 attack. In fact, the proposed algorithm selects Φ_j such that when embedded as the watermark vector and the MP3 compression-decompression attack is applied, the watermark recovery algorithm finds the correct watermark sign. This idea bears some similarity to the analysis-by-synthesis framework in speech coding in the codebook search for the best excitation.

The proposed algorithm works as follows: For each new audio frame it first selects the atom with the smallest dot product with the audio frame $X_i^T . \Phi_j$ (e.g., most orthogonal), and uses it for initial embedding. A super-frame (SF) is constructed which has the watermarked frame in the middle with the preceding 60ms and the following 60ms concatenated to it. This SF goes through the MP3 compression-decompression attack then the watermarked frame in the middle, which is now distorted, is extracted. The watermark recovery algorithm is applied. If the sign is correct the next audio frame is processed. If not, an exhaustive search over the L atoms is done, whereby each atom is embedded, the SF is created and the MP3 attack is applied.

The algorithm selects the atom which produces the correct sign and the highest ratio R_k between the largest correct sign and the largest wrong sign. If none of the L atoms produced a solution, the search is repeated once with an increase in embedding strength $\alpha_i' = 1.5 * \alpha_i$.

The algorithm continues until all frames have been watermarked. The algorithm steps are shown in table 2 below.

4.1 Adaptive Watermark Atom Selection Algorithm Steps

Input: Frame X_i, dictionary $\psi = [\Omega\ \Phi\ ]$
for i=1:I (frames)
Calculate α_i as in (15)
$\Phi = [\Phi_1\ \Phi_2\ \Phi_3 \ldots\ldots\ \Phi_j \ldots\ldots \Phi_{J-1}\ \Phi_J\ ]$ (Φ_j column vector atoms)
Find the column vector Φ_j for which $X_i^T . \Phi_j$ is minimum
Embed: $X_i^w = X_i + \alpha_i * \beta_k * \Phi_j$ (β_k=±1 = $\boldsymbol{w_i}$)
Create super-frame $X_i^s = [X_{i-m} \ldots . X_i^w \ldots . X_{i-m}\ ]$
Apply the MP3 compression on X_i^s
Apply MP3 decompression and extract the watermarked frame: $X_i^{w\prime}$
Solve the BPDN to get the watermark sparse vector β
Find the largest element in β

$$l = argmax\big(abs(\beta)\big)$$
$$w^1 = sign(\beta(l))$$

$if\ w^1 = \boldsymbol{w_i}$ Go to 1
else {
for j=1:L (Atoms in Φ)
for each Φ_j apply steps from (5-9), store the sign w_j^1 and the ratio R_j
Exclude the ones with wrong sign, sort the ones with the correct sign and select Φ_j with the largest R_j
End loop over j. if an atom is found Go to 1
else{
Do steps (11 to 14) with $\alpha_i' = 1.5 * \alpha_i$
end
end
end

5 Related Work in CS Watermarking

In 2007 Shiekh and Baraniuk [17] proposed a transform domain watermarking model based on compressed sensing for image watermarking assuming the host signal to be sparse in the DCT domain and the watermark was a randomly spread binary sequence. The work proposed here, in contrast to their work, assumes that the watermark is sparse and that the host can be sparsely coded by an overcomplete dictionary. In 2009 Tagliasacchi et. al. [18, 19] proposed a hash based tampering detection algorithm for audio and image data. If the tampering is sparse enough, it can be localized by solving the CS decoding problem. Their work is more oriented towards content authentication while the work here is focusing on robust watermarking. In [20, 21], the author of this paper has proposed a compressed sensing robust watermarking technique and compared it to the technique in [17]. In [20, 21], the watermark random vector was fixed and not adaptive. Thus, robustness of watermark was obtained however, with relatively low embedding rate that did not exceed 11bps for 64kbps MP3.

6 Experimental Results

The proposed algorithm was tested on parts of 3 MP3 songs used in [1, 20, 21], with 60 seconds from each, and the averaged results are shown here. The original MP3 are 128kbps mono, sampled at 44,100Hz. The MP3 attacks on the watermarked audio are 64kbps and 128kbps, and the payloads used are 689bps, 1378bps and 2756bps corresponding to audio frames of length 64, 32 and 16 samples respectively. The signal part of the dictionary is a concatenation of three inverse transforms each is M×M, namely, discrete cosine transform, Walsh-Hadamard transform and a Karhunen-Loeve transform trained on 60 seconds drawn from the 3 songs. Since the audio files are originally in MP3 128kbps format, each file is converted to the "WAV" format to do the watermark embedding, then the MP3 attack is applied either with a 128kbps or 64kbps compression. In the latter case, we get half the number of original samples of the 128kbps MP3 original file and an interpolation of order 2 is used to get the same number of samples again.

Figures 2, 3 and 4 summarize the results for the three payloads tested in the 64kbps MP3 attack, where the "No MP3" and "MP3" attack cases are with the chosen watermark atom that is most orthogonal to the audio frame. The "Proposed" is with the adaptive atom selection algorithm. Figures 5, 6 and 7 correspond to the 128kbps MP3 attack for the three payloads tested. Tables 2 and 3 summarize the best obtained results. The results show that high payloads of 689bps, 1378bps and 2756bps with acceptable %BER are obtained using the proposed algorithm with the 128kbps and 64kbps MP3 attacks. Without the proposed algorithm, the %Success is around (50%-65%), in the 64kbps, and (55%-80%) in the 128kbps MP3 attacks respectively, and gets worse with increasing L since more atoms are competing in the decoding stage. This increase in L becomes an advantage for the proposed algorithm since more L gives the algorithm a larger search space to find the best atom which is evident from figures 2 to 7 where the %Success approaches 100% with increasing *L*. It is to be

noted that the proposed algorithm pushes the complexity to the encoder side due to the best atom search phase. The algorithm complexity depends on the %success rate of the baseline watermark, and also depends on the number of atoms in the dictionary L, and on the complexity of the BPDN algorithm used.

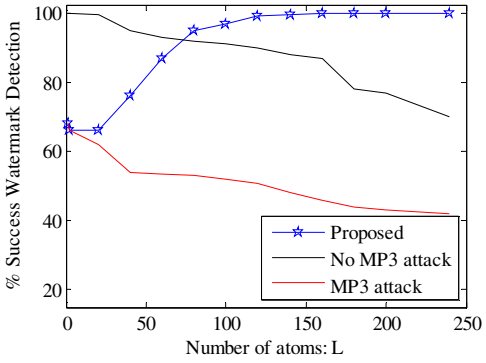


Fig. 2. M=64 (Payload 689bps), 64kbps MP3

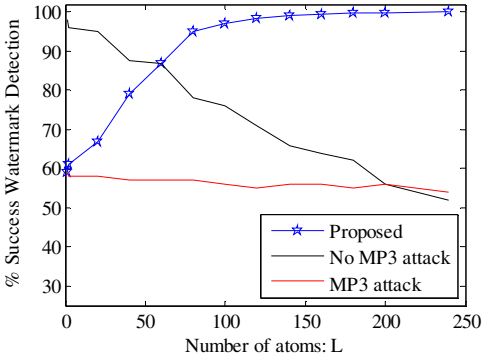


Fig. 3. M=32 (Payload 1378bps), 64kbps MP3

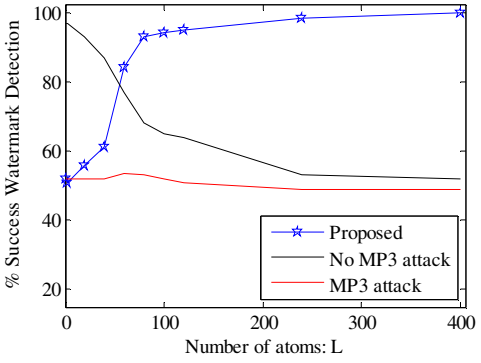


Fig. 4. M=16 (Payload 2756bps), 64kbps MP3

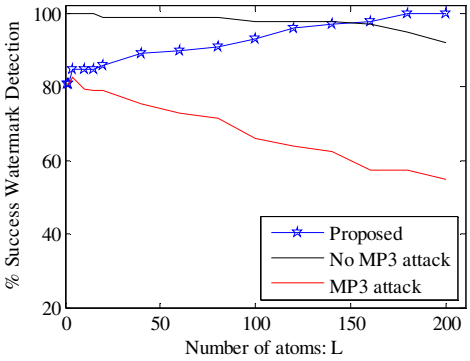


Fig. 5. M=64 (Payload 689bps), 128kbps MP3

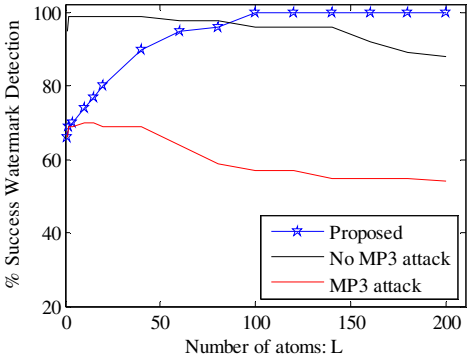


Fig. 6. M=32 (Payload 1378bps), 128kbps MP3

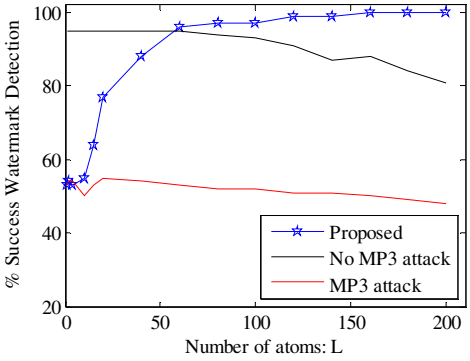


Fig. 7. M=16 (Payload 2756bps), 128kbps MP3

Table 2. Proposed Algorithm Results with 64kbps MP3 attack

MP3 Quality	SNR	%BER	Frame length	Payload	#Atoms
64kbps	30dB	0.3%	64 samples	689 bps	240
64kbps	30dB	0.5%	32 samples	1378 bps	240
64kbps	27dB	1.0%	16 samples	2756 bps	400

Table 3. Proposed Algorithm Results with 128kbps MP3 attack

MP3 Quality	SNR	%BER	Frame length	Payload	#Atoms
128kbps	30dB	0.1%	64 samples	689 bps	200
128kbps	30dB	0.3%	32 samples	1378 bps	200
128kbps	28dB	0.5%	16 samples	2756 bps	200

7 Conclusion and Future Work

A compressed sensing based, sparse coding watermarking framework is proposed where the watermark is a sparse vector with one non-zero element that takes the required sign of the encoded bit. The sparse coding technique uses an overcomplete dictionary that is a concatenation of a signal dictionary and a random Gaussian dictionary. The random dictionary is used to map the sparse watermark vector to a random watermark embedding vector. An adaptive watermarking algorithm is proposed where an atom from the random dictionary is selected for each new audio frame such that the (BPDN) decoding is robust to a single MP3 compression-decompression attack. The proposed technique is tested on three songs where the watermarked signals went through 64kbps and 128kbps MP3 attacks and the signal to embedding noise ratio (SNR) was kept above 27dB in all cases, and the watermarking effect was negligible.

High payloads (689, 1378 and 2756 bps) are achieved with acceptable quality and bit error rates. It is to be noted that the proposed algorithm is robust only to a single MP3 compression-decompression attack. Experimental results showed that repeating the attack degrades the watermark detection significantly. Also, the proposed algorithm was not tested under other attacks such as additive noise, synchronization, collusion, cropping/swapping and time-scale modification attacks.

Future work includes making the proposed algorithm robust to multiple MP3 compression-decompression attacks, and to other types of attacks. One direction is to synthesize a watermark random vector for each frame using a multi-objective optimization algorithm, where the watermark random vector is learned for each new frame. Another direction is to have atoms with different spectral characteristics, and choose the one that is least affected by the spectral distortion suffered by the MP3 attack. Finally, the proposed framework can be directly applied to Image and video watermarking with the emphasis on the robustness to JPEG and different video compression standards.

References

1. Noriega, R.M., Nakano, M., Kurkoski, B., Yamaguchi, K.: High Payload Audio Watermarking: toward Channel Characterization of MP3 Compression. Journal of Information Hiding and Multimedia Signal Processing 2(2), 91–107 (2011)
2. Vivekananda, B.K., Indranil, S., Abhijit, D.: An Audio Watermarking Scheme using Singular Value Decomposition and Dither-Modulation Quantization. Multimedia Tools and Applications Journal 52(2-3), 369–383 (2011)
3. Dhavale, S.V., Deodhar, R.S., Patnaik, L.M.: Walsh Hadamard Transform Based Blind Watermarking for Digital Audio Copyright Protection. In: Das, V.V., Thankachan, N. (eds.) CIIT 2011. CCIS, vol. 250, pp. 469–475. Springer, Heidelberg (2011)
4. Yang, H., Bao, D., Wang, X., Niu, P.: A Robust Content Based Audio Watermarking using UDWT and Invariant Histogram. Multimedia Tools and Applications Journal (November 2010)
5. El Hamdouni N., Adib A., Labri S., Torki M.: A Blind Digital Audio Watermarking Scheme Based on EMD and UISA Techniques. Multimedia Tools and Applications Journal (January 2012)
6. Tewari, T.K., Saxena, V., Gupta, J.P.: Audio Watermarking: Current State of Art and Future Objectives. International Journal of Digital Content Technology and Applications 5(7), 306–313 (2011)
7. Datta, K., Gupta, I.S.: Partial Encryption and Watermarking Scheme for Audio Files with Controlled Degradation of Quality. Multimedia Tools and Applications, Journal (2012)
8. Ercelebi, E., Batakci, L.: Audio watermarking Scheme Based on Embedding Strategy in Low Frequency Components with a Binary Image. Digital Signal Processing 19(2), 265–277 (2009)
9. Orsdemir, A., Altun, H.O., Sharma, G., Bocko, M.F.: On the Security and Robustness of Encryption via Compressed Sensing. In: IEEE Military Communication Conference MILCOM 2008, pp. 1–7 (2008)
10. Candès, E., Tao, T.: Decoding by Linear Programming. IEEE Transaction on Information Theory 51(12), 4203–4215 (2005)
11. Candès, E., Randall, P.: Highly Robust Error Correction by Convex Programming. IEEE Transaction on Information Theory 54(7) (2006)
12. Laska, J., Davenport, M., Baraniuk, R.: Exact Signal Recovery from Sparsely Corrupted Measurements through the Pursuit of Justice. In: Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, California (2009)
13. L1 magic: `http://users.ece.gatech.edu/~justin/l1magic/`
14. Gemmeke, J.F., Virtanen, T., Hurmalainen, A.: Examplar Based Sparse Representations for Noise Robust Automatic Speech Recognition. IEEE Trans. Audio, Speech and Language Processing 19(9), 2067–2080 (2011)
15. Sprechman, P., Sapiro, G.: Dictionary Learning and Sparse Coding for Unsupervised Clustering. In: ICASSP 2010, pp. 2042–2045 (2010)
16. Wright, J., Yi, M., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse Representation for Computer Vision and Pattern Recognition. Proc. of IEEE 98(6), 1031–1044 (2010)
17. Sheikh, M., Baraniuk, R.: Blind Error-Free Detection of Transform-Domain Watermarks. In: IEEE Int. Conf. on Image Processing (ICIP), San Antonio, Texas, vol. 5, pp. V-453–V-456 (September 2007)

18. Tagliasacchi, M., Valenzise, G., Tubaro, S.: Hash-Based Identification of Sparse Image Tampering. IEEE Transactions on Image Processing 18(11), 2491–2504 (2009)
19. Valenzise, G., Prandi, G., Tagliasacchi, M., Sarti, A.: Identification of Sparse Audio Tampering using Distributed Source Coding and Compressive Sensing Techniques. Eurasip Journal on Image and Video Processing 2009, 1–13 (2009)
20. Fakhr, M.W.: Robust Watermarking using Compressed Sensing Framework with Application to MP3. International Journal of Multimedia and its Applications, IJMA 4(6), 27–43 (2012)
21. Fakhr, M.W.: Sparse Watermark Embedding and Recovery using Compressed Sensing Framework for Audio Signals. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, China, pp. 535–539 (2012)

An HMM-Based Reputation Model

Ehab ElSalamouny[1,2] and Vladimiro Sassone[3]

[1] INRIA, France
[2] Faculty of Computer and Information Science, Suez Canal University, Egypt
[3] ECS, University of Southampton, UK

Abstract. In modern global networks, principals usually have incomplete information about each other. Therefore trust and reputation frameworks have been recently adopted to maximise the security level by basing decision making on estimated trust values for network peers. Existing models for trust and reputation have ignored dynamic behaviours, or introduced ad hoc solutions. In this paper, we introduce the HMM-based reputation model for network principals, where the dynamic behaviour of each one is represented by a hidden Markov model (HMM). We describe the elements of this novel reputation model. In particular we detail the representation of reputation reports. We also describe a mixing scheme that efficiently approximates the behaviour of a trustee given multiple reports about it from different sources.

1 Introduction

In modern global networks, the principals have incomplete information about each other. Therefore it is a challenging problem for a principal to take decisions regarding interactions with others. One approach that has recently adopted is to base these decisions upon a level of 'trust' associated with each network peer. We consider specifically the *probabilistic trust*, where the trust in a peer (trustee) *te* is expressed as a probability distribution over the potential outcomes of an interaction with *te*.

Many systems (e.g. [14,4,19,5]) have adopted the so-called *Beta model* [12], where the behaviour of a trustee *te* is modelled by a fixed rating θ approximating the probability that an interaction with *te* yields 'success'. This probability is learnt from past interactions with the trustee *te*. One limitation of the Beta-based systems is that they assume a fixed probabilistic behaviour for each principal; that is for each principal, there exists a fixed probability distribution over possible outcomes of its interactions. This assumption is indeed not realistic in practice. For example the behaviour of a principal can be significantly different when it is corrupted by an attacker.

As a step forward to handling dynamic behaviours, several papers, e.g. [12,4,20], adopted the 'decay' principle which was first introduced in [12]. It aims at capturing the recent behaviour of the trustee by letting older observations 'decay', so as to give higher weight to recent interactions over older ones. However, we showed in [7] that the decay principle is only useful when the trustee's behaviour is highly *stable*, i.e, when the probability distribution over the observables is unlikely to change.

For coping with this limitation, we introduced in [9] the foundations of a novel HMM-based trust model to evaluate trust in trustees exhibiting dynamic behaviours.

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 111–121, 2013.

A trustee is characterised by a set of (behavioural) states, each associated with a probability distribution over observable outcomes of interactions. It proceeds by performing (unobservable) probabilistic state transitions, which in turn determine changes in the statistical properties of the (observable) outcomes of interactions. These assumptions are met exactly by representing the trustee with a finite-state hidden Markov model (HMM) [18], called the 'real' model. In particular, the trustee's state transitions are hidden, and trusters observe only the outcomes of their interactions with it.

For evaluating the trust, past observations are used to learn the trustee's behaviour, and then predict the outcomes of future interactions. The key information for that is the *real* predictive distribution, i.e., the probability of each potential outcome in the next interaction between a truster and a trustee *te*, given the outcomes of past interactions. Yet, since the real model λ for *te* is unknown, the truster can only estimate the real predictive distribution. In our approach this is done using the *Baum-Welch* algorithm [18] that yields an 'approximation' η of λ, and then use η to evaluate the so-called *estimated* predictive distribution, which ultimately defines the truster's trust in *te*.

In many cases, the sequence of direct observations available to the truster is not sufficiently long to learn the behaviour of the trustee with a satisfactory accuracy. To handle this shortage of information, the *reputation* information is used in the learning process. This information is seen as *reports* - about the trustee - given by other principals called *reputation sources*. These reports enrich the truster's knowledge about the trustee and therefore enhance the approximation of λ. This also implies a better estimate of the predictive distribution. Thus the notion of reputation raises two main questions:

- Which representation is appropriate for a reputation report ?
- How are reports, from different sources, utilised to enhance estimations ?

Clearly, the answers depend on the assumptions made about the real model of the trustee's behaviour. For example, in systems based on the Beta reputation model and its extensions (e.g., [12,11,15]), a reputation report consists of the count of each outcome experienced by its source in its interactions with the trustee. Multiple reputation reports from different sources are therefore mixed by adding up the corresponding counts of outcomes. This is not so easy in our case, where we must take into account that observations seen by different sources could correspond to different (hidden) states of the trustee, and can not therefore be summed together. In fact, this is a major technical challenge we face, one which demands a new approach.

In this paper we introduce a reputation model that matches the 'dynamic' nature of the trustee's behaviour. This model answers the above two questions when the behaviour is represented by an HMM, and therefore completes the basic HMM-based trust model of [9] with a reputation handling mechanism. We also point to experimental evidence, through simulations, in support of our model. To the best of our knowledge, this is the first trust-and-reputation model for multi-state, dynamic systems. Thus, it provides the first complete answer to the research challenge launched in [15].

Structure of the paper. The next section concisely introduces hidden Markov models. In Section 3 we recall the basic model of HMM-based trust, whilst in Section 4 we detail the elements of our HMM-based reputation model. Section 5 sketches an experimental analysis of our reputation model against some of its predecessors. Finally, we conclude

our results in Section 6. For space restrictions, the proofs are omitted from the body of the paper. The interested reader can find them in the online version [8].

2 Hidden Markov Models (HMMs)

A *Hidden Markov Model (HMM)* [1] is a probabilistic model essentially based on a notion of system state. Underlying any HMM there is a Markov chain modelling (probabilistically) system's transitions between a set of states. Each state in this chain is associated with a particular probability distribution over the set of possible outcomes (observables). The output of an HMM is a sequence of outcomes where each outcome is sampled according to the probability distribution of the underlying state. In the following, we denote the state of the HMM and the outcome at time t by q_t and o_t respectively.

Definition 1 (hidden Markov model). A (discrete) *hidden Markov model* (HMM) is a tuple $\lambda = (S, V, \boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ where $S = \{1, 2, \ldots, N\}$ is a finite set of *states*; $V = \{z_1, z_2, \ldots, z_K\}$ is a finite set of possible *observables*; $\boldsymbol{\pi}$ is a distribution on S, the *initial distribution*; $\boldsymbol{A} : S \times S \to [0, 1]$ is the *state transition matrix*, with $A_{ij} = P(q_{t+1} = j \mid q_t = i)$ and $\sum_{j \in S} A_{ij} = 1$; and $\boldsymbol{B} : S \times V \to [0, 1]$ is the *emission matrix*, with $B_i(z_k) = P(o_t = z_k \mid q_t = i)$, $\sum_{z_k \in V} B_i(z_k) = 1$.

As an example, Figure 1 shows a two-state HMM with the observation set $\{\mathsf{s}, \mathsf{f}\}$, where transitions $\mathbf{1} \mapsto \mathbf{2}$ and $\mathbf{2} \mapsto \mathbf{1}$ have probabilities 0.1, 0.12 respectively. The probabilities of self-transitions $\mathbf{1} \mapsto \mathbf{1}$ and $\mathbf{2} \mapsto \mathbf{2}$ are therefore 0.9, 0.88 respectively. The other parameters $\boldsymbol{\pi}, \boldsymbol{B}$ are shown.

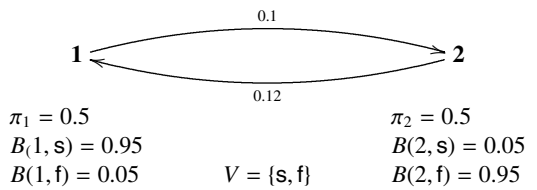


Fig. 1. Example Hidden Markov Model

HMMs provide the computational trust community with several obvious advantages: they are widely used in scientific applications, and come equipped with efficient algorithms for computing the probabilities of events and for parameter estimation [18]. It is worth noticing that HMMs are a generalisation of the Beta model [12] to a multi-state model. In fact, representing the behaviour of a trustee *te* by an HMM λ provides, for each state j of *te*, a distribution $\boldsymbol{B}_j$ over candidate observables V.

According to Definition 1 of HMM, the probability of a sequence of outcomes $h = o_1\, o_2 \cdots o_T$ given an HMM λ is given by the following equation.

$$P(h \mid \lambda) = \sum_{q_1, \ldots, q_T \in S} \pi(q_1) \cdot B_{q_1}(o_1) \cdot A_{q_1 q_2} \cdot B_{q_2}(o_2) \cdots A_{q_{T-1} q_T} \cdot B_{q_T}(o_T). \qquad (1)$$

The above probability is evaluated efficiently by the *forward-backward* algorithm ([18]), which evaluates the above probability inductively on the time t where $1 \leq t \leq T$.

In our work we assume that utilised HMMs are *ergodic*. This corresponds to demanding that the Markov chain underlying an HMM is irreducible and aperiodic (c.f. [10,16,3]). An HMM is *irreducible* if each state is reachable (with non-zero probability) from every other one. It is *aperiodic* if it has at least one aperiodic state. A state i is aperiodic if it does not recur with a cyclic period, that is if the greatest common divisor of times $t > 0$ such that $P(q_{1+t} = i \mid q_1 = i) > 0$ is 1. To guarantee aperiodicity it is sufficient that one state has a self-transition with non-zero probability. Such conditions are not overly restrictive for practical systems and typical applications.

2.1 Baum-Welch Algorithm

Given a fixed set S of states and a fixed set V of observables, the Baum-Welch (BW) algorithm [2,18] iteratively finds an HMM η which maximises the probability $P(h \mid \eta)$ of a given sequence h. This iterative algorithm starts with an initial HMM η' having parameters $\boldsymbol{\pi}'$, $\boldsymbol{A}'$, and $\boldsymbol{B}'$. Then at each iteration, the a priori HMM η' is refined to obtain a posteriori HMM η with parameters $\boldsymbol{\pi}$, $\boldsymbol{A}$, and $\boldsymbol{B}$. These parameters are evaluated by mathematical equations which we here summarise only informally by means of Equations (2), (3), and (4), respectively.

$$\pi_i = \text{the probability of being in state } i \text{ at time } (t = 1). \tag{2}$$

$$A_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}. \tag{3}$$

$$B_i(z_k) = \frac{\text{expected number of times in state } i \text{ and observing symbol } z_k}{\text{expected number of times in state } i}. \tag{4}$$

In the above equations, the expected values are computed given the sequence h of observations, and the probability distributions defined by the a priori model η'. The resulting a posteriori model becomes the a priori one for the next iteration. The algorithm stops when the a priori and a posteriori models have the same parameters. More details about this algorithm can be found in [18]. One limitation of this algorithm is that it only converges to a local maxima for the probability function rather than the global one.

3 Hmm-Based Trust Model

The *HMM-based trust model* [9] is based on the assumption that the behaviour of the trustee *te* is dynamic. This 'unknown' behaviour is modelled by an HMM λ, called the 'real' model of *te*. Since λ is generally unknown, each truster *tr* approximates it by a finite-states HMM η, which we call the 'approximate' model of *te*.

In this approximation (learning) process, the truster *tr* uses past outcomes of its direct interactions with *te* as follows. Given a sequence h of outcomes of direct interactions between *tr* and *te*, the truster *tr* applies the BW-algorithm [2,18] to h. As described earlier, this algorithm yields an HMM that maximises the probability of h, and therefore

defines the required approximate model η of *te*. We remark here that the size of η is fixed by the truster, and represents the approximation level of the trustee's behaviour.

Using the in-hand approximate model η for *te*, the truster *tr* can estimate a probability distribution over possible outcomes of its next interaction with *te*. This distribution, is called an 'estimated' predictive distribution of *te* (from *tr*'s point of view), and represents the trust of *tr* in *te*. Actually, this distribution is meant to be an 'estimate' for the 'real' predictive distribution which is determined by the real model λ of the trustee. The quality of this estimation is quantified by the difference between the 'real' and 'estimated' predictive distributions as follows.

Estimation error. The Kullback-Leibler (KL) divergence [13,6] is an appropriate measure for the difference between distributions. It has an information-theoretic flavour, and is technically understood as a measure for the lost information when a probability distribution is approximated by another. In our case, we write the real and estimated predictive distributions as $P(.)$ and $\mathcal{H}(.)$ respectively. The KL-divergence from $P(.)$ to $\mathcal{H}(.)$ is called the 'estimation error' and is defined as follows.

$$D_{KL}(P(.) \parallel \mathcal{H}(.)) = \sum_{z \in V} P(z) \log\left(\frac{P(z)}{\mathcal{H}(z)}\right). \tag{5}$$

Note that the above estimation error is specific to a pair of distributions corresponding to a particular sequence of outcomes. Thus, we evaluate the quality of our trust model as the *average* over all possible sequences. This average is the expected value of above divergence, and therefore is referred to as the 'expected estimation error'.

4 Hmm-Reputation Model

Now we describe our proposed reputation model, which enhances the trust evaluation using supplementary feedback reports about the trustee. This model consists of two main components: a formalism of reputation reports exchanged between the network peers; and a mixing scheme which uses multiple reputation reports about a trustee *te* to evaluate the trust in *te*. As this trust is an estimated predictive distribution, our goal is to design those components such that the expected estimation error, described in Section 3 is minimised.

We start by linking the expected estimation error (for a trustee *te*) to the sequences of outcomes observed by all reputation sources. Let h be a sequence of outcomes resulting from past interactions between a trustee *te* and a single reputation source. Let also λ be the unknown real model of *te*. It is shown in [9] that with any approximate model η, the expected estimation error converges (as the length T of h grows) to the following limit which we refer to as the *asymptotic estimation error*.

$$Error\left(\lambda, \mathcal{H}_\eta\right) = C(\lambda) - H(\lambda, \eta), \tag{6}$$

where $C(\lambda) = \lim_{T\to\infty} \mathbf{E}\left[\log P(o_{T+1} \mid q_T, \lambda)\right]$, $H(\lambda, \eta) = \lim_{T\to\infty} \mathbf{E}\left[\log P(o_{T+1} \mid h, \eta)\right]$. In the language of information theory, $-C(\lambda)$ is the expected *entropy* of the real predictive distribution determined by λ (the real model); and $-H(\lambda, \eta)$ is the expected

cross-entropy between the real and estimated predictive distributions where the latter is determined by both λ and η. By the asymptotic properties of ergodic HMMs [1, Theorem 3.2], the log probability of any T-length observation sequence h, generated by λ, is related to $H(\lambda, \eta)$ as follows.

$$(1/T) \log P(h \mid \eta) \overset{a.s.}{\to} H(\lambda, \eta), \tag{7}$$

that is, the left-hand term converges *almost surely* (with probability 1) to $H(\lambda, \eta)$ as $T \to \infty$. Now consider the interactions between a trustee *te* and a set of reputation sources $\mathcal{M} = \{1, 2, \ldots, M\}$ until a certain time instant. For every reputation source $u \in \mathcal{M}$, let h^u be the sequence of outcomes observed by u. Let also T_u be the length of h^u. By Eq. (7), the average of the quantities $(1/T_u) \log P(h^u \mid \eta)$ over the elements of $\mathcal{M}$ approximates $H(\lambda, \eta)$. Thus, minimising the asymptotic estimation error, expressed by (6), amounts to choosing the approximate model η that maximises such an average, i.e. choosing the approximate model η that satisfies the following equation.

$$\eta = \underset{\mathcal{R}_n}{\operatorname{argmax}}\, \mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \mathcal{R}_n\right), \tag{8}$$

where $\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \mathcal{R}_n\right)$ is an objective function whose value depends on an n-state HMM $\mathcal{R}_n$. This objective function is defined as

$$\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \mathcal{R}_n\right) = \sum_{u \in \mathcal{M}} (1/T_u) \log P(h^u \mid \mathcal{R}_n). \tag{9}$$

Maximising the above objective function requires full access to all the sequences h^u. However, it is not practical for principals to exchange their entire observed sequences because each one of these sequences gets longer over time. Therefore we alternatively maximise a tight lower bound for this function. For doing so, we assume that each reputation source u uses its own observation sequence h^u to learn an 'a priori' approximate HMM η^u for the trustee. In terms of the a priori HMMs and other variables, the following lemma provides the required lower bound for the objective function.

Lemma 1. *Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be a set of reputation sources. For all $u \in \mathcal{M}$, let h^u and $\eta^u = (S, V, \boldsymbol{\pi}^u, \boldsymbol{A}^u, \boldsymbol{B}^u)$ be, respectively, the sequence of outcomes observed by u, and the corresponding a priori HMM. For any sequence q of states, let $P(q \mid h^u, \eta^u)$ denote the probability of q given h^u and η^u. Thus, it holds for every a posteriori HMM $\eta = (S, V, \boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ that*

$$\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \eta\right) \geq \sum_{u \in \mathcal{M}} (1/T_u)\, \mathcal{Q}(\eta^u, h^u, \eta) + \sum_{u \in \mathcal{M}} (1/T_u)\, \mathcal{R}(\eta^u, h^u),$$

$$\textit{where} \qquad \mathcal{Q}(\eta^u, h^u, \eta) = \sum_q P(q \mid h^u, \eta^u) \log P(h^u, q \mid \eta), \tag{10}$$

$$\mathcal{R}(\eta^u, h^u) = -\sum_q P(q \mid h^u, \eta^u) \log P(q \mid h^u, \eta^u); \tag{11}$$

and the equality holds when $\eta = \eta^1 = \eta^2 = \cdots = \eta^M$.

Lemma 1 provides a lower bound for $\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \eta\right)$ given any a posteriori model η. Note that this bound is tight in the sense that it is equal to the objective function when the a priori models η^u are all equal to the a posteriori model η. Thus, we set our objective in the following to compute the *optimal* a posteriori model η^* which we define as the one maximising the above lower bound. That is,

$$\eta^* = \underset{\eta}{\operatorname{argmax}} \left(\sum_{u \in \mathcal{M}} (1/T_u)\, \mathcal{Q}(\eta^u, h^u, \eta) + \sum_{u \in \mathcal{M}} (1/T_u)\, \mathcal{R}(\eta^u, h^u) \right). \tag{12}$$

We will show that a truster wanting to compute η^* needs to collect only certain statistics about every observation sequence h^u, rather than the entire sequence. For fixed sets of states S and observables V, let the sequence of outcomes observed by the reputation source u be $h^u = o_1^u\, o_2^u \ldots o_{T_u}^u$, where o_t^u is the outcome at time t. Similarly, let $q^u = q_1^u\, q_2^u \ldots q_{T_u}^u$ be the (hidden) sequence of states underlying the observed sequence h^u. The optimal a posteriori HMM η^* in (12) is computed by the following theorem.

Theorem 1. *Given a set $\mathcal{M}$ of reputation sources, the parameters of the optimal a posteriori HMM $\eta^* = (S, V, \boldsymbol{\pi}^*, \boldsymbol{A}^*, \boldsymbol{B}^*)$, are given by the following equations.*

$$\pi_i^* = \frac{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} P\left(q_1^u = i \mid h^u, \eta^u\right)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u}}, \tag{13}$$

$$A_{ij}^* = \frac{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i, q_t^u = j \mid h^u, \eta^u\right)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i \mid h^u, \eta^u\right)}, \tag{14}$$

$$B_i^*(z_k) = \frac{\sum\limits_{u \in \mathcal{M}} \sum_{t=1,\, o_t^u = z_k}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=1}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right)}. \tag{15}$$

In the context of the BW-algorithm [18,17], $P(q_t = i \mid h, \eta)$, the probability of visiting state i at time t given an observation sequence h and an HMM η is denoted by the variable $\gamma_t(i)$. Also $P(q_{t-1} = i, q_t = j \mid h, \eta)$, the probability of visiting states i and j at times $t-1$ and t respectively is denoted by the variable $\xi_{t-1}(i, j)$. In the same manner, we use the variables $\gamma_t^u(i)$ and $\xi_{t-1}^u(i, j)$ to denote the probabilities $P\left(q_t^u = i \mid h^u, \eta^u\right)$ and $P\left(q_{t-1}^u = i, q_t^u = j \mid h^u, \eta^u\right)$, respectively. Using these variables, (13-15) can be written in shorter forms as follows.

$$\pi_i^* = \frac{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \gamma_1^u(i)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u}}, \quad A_{ij}^* = \frac{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=2}^{T_u} \xi_{t-1}^u(i, j)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=2}^{T_u} \gamma_{t-1}^u(i)}, \quad B_i^*(z_k) = \frac{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=1,\, o_t^u = z_k}^{T_u} \gamma_t^u(i)}{\sum\limits_{u \in \mathcal{M}} \frac{1}{T_u} \sum_{t=1}^{T_u} \gamma_t^u(i)}.$$

From these formulae we notice that the computation of η^* does not require full knowledge about the observation sequences h^u, where $u \in \mathcal{M}$, but only some statistical functions of these sequences computed 'locally' in individual reputation sources.

$$\bar{\gamma}_1^u = \left[\, \bar{\gamma}_1^u(1)\ \bar{\gamma}_1^u(2)\ \ldots\ \bar{\gamma}_1^u(N) \,\right], \qquad \text{where} \quad \bar{\gamma}_1^u(i) = \frac{1}{T_u}\,\gamma_1^u(i)$$

$$\bar{\gamma}_{T_u}^u = \left[\, \bar{\gamma}_{T_u}^u(1)\ \bar{\gamma}_{T_u}^u(2)\ \ldots\ \bar{\gamma}_{T_u}^u(N) \,\right], \qquad \text{where} \quad \bar{\gamma}_{T_u}^u(i) = \frac{1}{T_u}\,\gamma_{T_u}^u(i)$$

$$\bar{\gamma}^u = \left[\, \bar{\gamma}^u(1)\ \bar{\gamma}^u(2)\ \ldots\ \bar{\gamma}^u(N) \,\right], \qquad \text{where} \quad \bar{\gamma}^u(i) = \frac{1}{T_u}\sum_{t=1}^{T_u-1}\gamma_t^u(i)$$

$$\bar{\xi}^u = \begin{bmatrix} \bar{\xi}^u(1,1) & \bar{\xi}^u(1,2) & \ldots & \bar{\xi}^u(1,N) \\ \bar{\xi}^u(2,1) & \bar{\xi}^u(2,2) & \ldots & \bar{\xi}^u(2,N) \\ \vdots & \ldots & \ddots & \vdots \\ \bar{\xi}^u(N,1) & \bar{\xi}^u(N,2) & \ldots & \bar{\xi}^u(N,N) \end{bmatrix}, \qquad \text{where} \quad \bar{\xi}^u(i,j) = \frac{1}{T_u}\sum_{t=2}^{T_u}\xi_{t-1}^u(i,j)$$

$$\bar{\omega}^u = \begin{bmatrix} \bar{\omega}^u(1,1) & \bar{\omega}^u(1,2) & \ldots & \bar{\omega}^u(1,K) \\ \bar{\omega}^u(2,1) & \bar{\omega}^u(2,2) & \ldots & \bar{\omega}^u(2,K) \\ \vdots & \ldots & \ddots & \vdots \\ \bar{\omega}^u(N,1) & \bar{\omega}^u(N,2) & \ldots & \bar{\omega}^u(N,K) \end{bmatrix}, \qquad \text{where} \quad \bar{\omega}^u(i,k) = \frac{1}{T_u}\sum_{t=1,\, o_t^u = z_k}^{T_u}\gamma_t^u(i)$$

Fig. 2. The elements of an HMM-based reputation report

It is essential to ensure that the a priori HMMs η^u have the same set S of states (as required by Lemma 1 and Theorem 1). Therefore we define a parameter $\bar{\eta}$ for the reputation protocol to be an initial N-state HMM. The parameter $\bar{\eta}$ is shared by all principals and is regarded as the 'default' trustee's behaviour which is refined through the learning process to η^u according to the sequence of outcomes h^u seen by u.

Thus, we describe the HMM based reputation model as follows. Using $\bar{\eta}$ as an initial HMM, each reputation source u applies the BW-algorithm to its own observations h^u about the trustee te. This learning process, performed by u, yields an HMM η^u approximating the behaviour of te (from u's point of view) and also the variables $\gamma_t^u(i)$, $\xi_t^u(i,j)$ for $1 \le t \le T_u$, and all $i, j \in \{1, 2, \ldots, N\}$. In terms of these variables, every reputation source u constructs its reputation report about te as the tuple $\left(T_u\,,\, \bar{\gamma}_1^u\,,\, \bar{\gamma}_{T_u}^u\,,\, \bar{\gamma}^u\,,\, \bar{\xi}^u\,,\, \bar{\omega}^u\right)$. While T_u is clearly the length of h^u, each other element in this tuple (report) is basically a matrix defined in Figure 2.

Now we describe our reputation mixing scheme. Suppose that multiple reports about te, from a set $\mathcal{M}$ of reputation sources, are available to a truster tr. Note that these reports include the one constructed by tr itself about te. Using the elements of these reports, tr computes the optimal a posteriori HMM η^* of te as follows

$$\pi_i^* = \frac{\sum_{u\in\mathcal{M}} \bar{\gamma}_1^u(i)}{\sum_{u\in\mathcal{M}} \frac{1}{T_u}}, \qquad A_{ij}^* = \frac{\sum_{u\in\mathcal{M}} \bar{\xi}^u(i,j)}{\sum_{u\in\mathcal{M}} \bar{\gamma}^u(i)}, \qquad B_i^*(z_k) = \frac{\sum_{u\in\mathcal{M}} \bar{\omega}^u(i,k)}{\sum_{u\in\mathcal{M}} \left(\bar{\gamma}^u(i) + \bar{\gamma}_{T_u}^u(i)\right)}.$$

Using η^* and the past observations h^{tr} (seen by *tr*), the trust of *tr* in *te* is evaluated as an estimated predictive distribution, i.e. the probability - given h^{tr} and η^* - of every possible outcome $z_k \in V$ for a new interaction with *te*. This distribution is expressed as

$$P(z_k \mid h^{tr}, \eta^*) = P(z_k, h^{tr} \mid \eta^*)/P(h^{tr} \mid \eta^*) \qquad \forall z_k \in V.$$

In the above equation, the probabilities on the right side are defined by Eq. 1, and efficiently evaluated by the *forward-backward* algorithm ([18]).

5 Experimental Evaluation

To evaluate our reputation model experimentally, we simulate an HMM λ representing a trustee *te*. We consider two partners: *tr* and *rs*. Each interaction between *te* and a partner is simulated by allowing λ to make a state transition, and produce an observation in one partner. Over time we allow the reputation source *rs* to serve a reputation report about *te* to the truster *tr*, which combines it with its local trust information to produce a new estimated predictive distribution for *te*. We then evaluate the expected estimation error, defined in Section 3, using Monte-Carlo approach (see, e.g., [3]).

Let λ be a 4-state HMM representing a 'stable' behaviour, where the probability of making self-transition is high (0.9), and other transitions are equally likely. Let the observation alphabet $V = \{1, 2\}$, where the emission matrix is

$$B_\lambda = \begin{bmatrix} 1.0 & 0.0 \\ 0.7 & 0.3 \\ 0.3 & 0.7 \\ 0.0 & 1.0 \end{bmatrix} \tag{16}$$

At each interaction with *te*, we assume the *tr* and *rs* are equally likely (with probability 0.2) to be *te*'s partner, while it remains a probability 0.6 that any other principal is the partner. Figure 3, shows the impact of using the HMM-based reputation model on the expected estimation error. The graph on the left compares the HMM trust model *with* and *without* reports from reputation sources. The higher curve shows the estimation error resulting from using only *tr*'s observations (1 report). The lower curve shows the error when *tr* uses also the reputation report collected from *rs* (2 reports). The improvement resulting from mixing in the reputation report is indicated by the vertical gap between the curves. Observe that this becomes less significant as the total number of interactions grows. This is because in the case of long sequences, the observations made individually by *tr* tend to be sufficient to learning the trustee's behaviour with accuracy.

With the same simulation framework, the right-hand side of Figure 3 compares the expected estimation errors of our model against the Beta reputation model. Observe that for a relatively low number of total interactions T, the beta model outperforms the HMM reputation model by exhibiting a lower expected estimation error. In this case, the combined length of the observation sequences is not sufficient to capture the 'dynamicity' of *te*'s behaviour. As a result, the learning algorithm working on such input produces a low-quality approximate behaviour HMM η. Hence the large estimation error compared to using Beta reputation reports. However, when the number of interactions grows and the sequence become long enough to compute a good estimate, the HMM reputation model exhibits a substantially lower error than the Beta model.

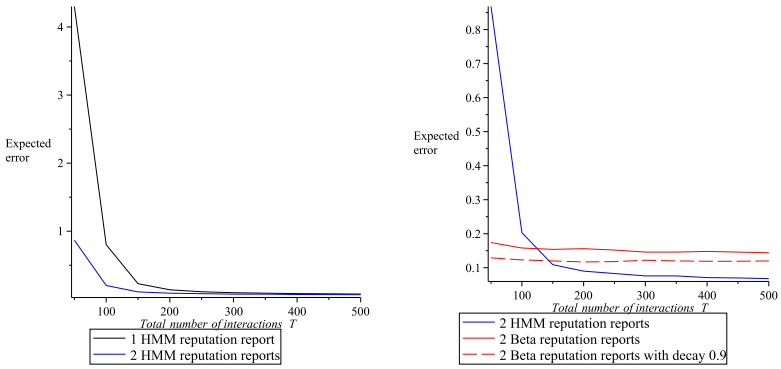


Fig. 3. The expected estimation error using the HMM-based reputation model

The Beta model falls short here: since it ignores *te*'s dynamic behaviour and only learns an 'average' probability distribution over possible outcomes, it can not keep the expected estimation level low. It is also apparent from Figure 3, that incorporating a decay factor in the Beta model (viz., 0.9 in the example) reduces the expected estimation error. Further details about the effect of decay can be found in [7].

6 Conclusion

We proposed a model for reputation which completes our basic HMM-based trust model in [9] and yields the first trust-and-reputation model for multi-state, dynamic systems. The reputation model enhances the quality and reliability of trust judgements and their evaluation process by using feedback information about the trustee in the form of *reputation reports*. The latter are a 'digest' of a principal's interactions with the trustee, conceptually nothing but an abstraction on the simple idea of ratings given by *reputation sources* about trustees. The model provides mixing equations which can be used by the truster to combine reputation reports collected from different sources, together with its own trust information, in order to evaluate its trust in the trustee.

We used the same experimental approach previously used in [9] to evaluate and compare trust models in terms of the expected estimation error. This allows us to investigate the impact of the HMM-reputation model on trust evaluation, as well as to compare this model against its predecessors. We found that the estimation error is significantly reduced when multiple reputation reports are used in the trust evaluation process. We also discussed how the improvement due to reputation reports gets less significant as the total number of interactions with the trustee gets larger. This is because a larger number of total interactions with the trustee implies that the single sequence experienced by the truster itself tends to provide a sufficient accurate basis to learn the trustee's behaviour.

A comparison with the Beta reputation model, using the same number of reputation sources, yielded that the Beta reputation model outperforms our HMM-based reputation when the total number T of interactions is relatively small. As T gets larger, the HMM-based reputation model gradually improves in terms of the estimation error,

and eventually outperforms the Beta model very significantly. This is because longer observation sequences imply more accurate approximate models of the trustee's dynamic behaviour, information which, in contrast, the Beta model ignores altogether.

References

1. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite-state Markov chains. Annals of Mathematical Statistics 37(6), 1554–1563 (1966)
2. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics 41(1), 164–171 (1970)
3. Brémaud, P.: Markov chains: Gibbs fields, Monte Carlo simulation, and queues. Springer (1998)
4. Buchegger, S., Le Boudec, J.: A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In: P2PEcon 2004 (2004)
5. Cahill, V., Gray, E., Seigneur, J.M., Jensen, C.D., Chen, Y., Shand, B., Dimmock, N., Twigg, A., Bacon, J., English, C., Wagealla, W., Terzis, S., Nixon, P., Serugendo, G., Bryce, C., Carbone, M., Krukow, K., Nielsen, M.: Using trust for secure collaboration in uncertain environments. IEEE Pervasive Computing 2(3), 52–61 (2003)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience (July 2006)
7. ElSalamouny, E., Krukow, K., Sassone, V.: An analysis of the exponential decay principle in probabilistic trust models. Theoretical Computer Science 410(41), 4067–4084 (2009)
8. ElSalamouny, E., Sassone, V.: An hmm-based reputation model. Technical report, INRIA (2013), `http://hal.inria.fr/hal-00831401`
9. ElSalamouny, E., Sassone, V., Nielsen, M.: HMM-based trust model. In: Degano, P., Guttman, J. (eds.) FAST 2009. LNCS, vol. 5983, pp. 21–35. Springer, Heidelberg (2010)
10. Grimmet, G., Stirzaker, D.: Probability and Random Processes, 3rd edn. Oxford University Press (2001)
11. Jøsang, A., Haller, J.: Dirichlet reputation systems. In: The Second International Conference on Availability, Reliability and Security, ARES 2007, pp. 112–119 (2007)
12. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings from the 15th Bled Conference on Electronic Commerce, Bled (2002)
13. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
14. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation (for ebusinesses). In: Proceedings from 5th Annual Hawaii International Conference on System Sciences, HICSS 2002, p. 188. IEEE (2002)
15. Nielsen, M., Krukow, K., Sassone, V.: A bayesian model for event-based trust. Electronic Notes in Theoretical Computer Science (ENTCS) 172, 499–521 (2007)
16. Norris, J.R.: Markov chains. Cambridge University Press (1997)
17. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. United states ed edn. Prentice Hall PTR (April 1993)
18. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
19. Teacy, W., Patel, J., Jennings, N., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12, 183–198 (2006)
20. Xiong, L., Liu, L.: PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)

Towards IT-Legal Framework for Cloud Computing

Sameh Hussein and Nashwa Abdelbaki

School of Communications and Information Technology, Nile University, Egypt
{sameh.hussein@nileu,nabdelbaki@nileuniversity}.edu.eg

Abstract. As the common understanding of Cloud Computing is continuously evolving, the terminology and concepts used to define it often need clarifying. Therefore, Cloud customers and Cloud Providers are used to dispute about Service Level Agreements, Service Level Objectives and Quality of Service. Simultaneously, SLAs/SLOs/QoS represent other related technical problems such as Security, Privacy, Compliancy and others. Technical problems are usually defined within technical context, where both parties ignore analyzing problem's legally related causes. In fact, these problems are stemming from the mapping and translating of surrounding laws and regulations into technical terms. In this paper we propose an IT based Legal Framework for Cloud Computing. It helps Cloud customers and Cloud Providers to avoid having such problems. Also it manages the interlocks in the gray area between the IT department and the legal department via setting clear boundaries, resolving conflict of interests and offering segregation of duties.

Keywords: Cloud Computing (CC), Legal Issues, Legal Framework, Privacy, Data Protection, Service Level Agreements/Objectives (SLA/SLO), Quality of Service (QoS).

1 Introduction

In the past, we had conventional data centers where the IT service providers offer only the infrastructure, and the administrative duties were at the customer side. Grid computing came after to offer high performance services. Nowadays, Cloud Computing (CC) becomes very essential need for companies and organizations, where cost cutting, transferring risks and other benefits take place. However, many legal and regulatory challenges appear and grow with the continuous migration towards Cloud technology. CC characteristics such as on demand self-service, elasticity, metered service or ubiquitous access make it looks simple, easy and Business as Usual (BaU) operation. CC terms and conditions are filled with traps as well as legal issues and challenges.

Companies are making their solutions available through CC without examining the characteristics, models and services involved. This leads to misunderstand what CC is and what it can become [1], [2]. However, the decision towards Cloud technology leads to many risks which are stemming from the dependability on third-party Clouds. As the more data or information clients transfer to Clouds, the more risks they experience [3], [4].

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 122–130, 2013.

Many governments and national entities prepare themselves towards Cloud technology. However, they have people private/confidential information where laws, regulations & restrictions vary from a country/region to another [5]. These entities suffer lack of comprehensive regulations and official standards. In this paper, we study the gray area challenges, restrictions and legal/technical pair of issues. Accordingly, we come-up with our proposed IT based Legal Framework for CC. Today's CC network, Fig 1, shows the difficulty to manage its many interlocks between Cloud Providers, customers and clients.

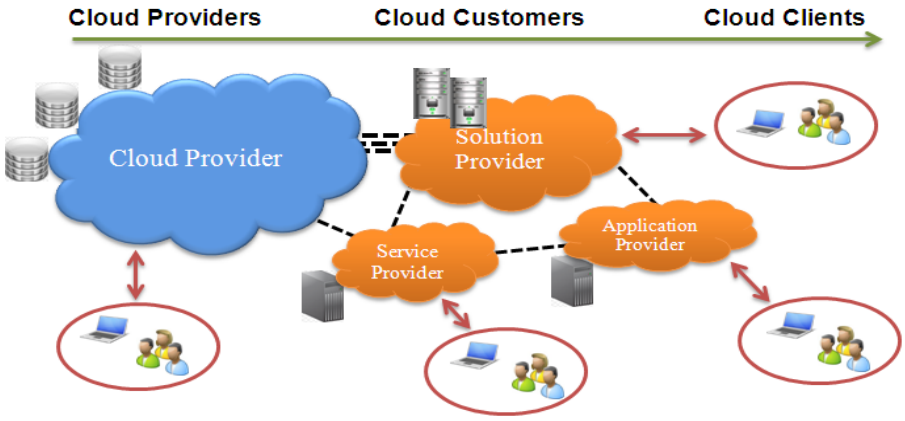


Fig. 1. Today's Cloud Computing Network

Through the rest of this paper, we define and address technical major problems caused by legal aspect, SLA, SLO and QoS in section 2. In addition to that, we deeply explain our proposed Legal Framework in section 3. Then we conclude and expect the future work in section 4.

2 Cloud Computing Legal Problems Definition

Legal issues are used to lead to many technical problems which have not been fully understood nor fully standardized within its main legal reason [6]. Usually, Cloud clients ignore the grey area between legal and IT departments. Many customers are doing their best to address & compensate these technical problems with their Cloud Providers (CPs). This is done without carefully studying the laws, regulations, certificates, commitments, and other binding contracts [7].

2.1 Problems Inter-related with Legal Issues

Contracting with CPs is very tough and complex. Consequently, legal issues/SLAs/SLOs/QoS cause major technical problems which come together with cloud benefits. We can represent that in a scenario (Fig 2) where CPs rain many legal and regulatory issues. They fall on customer territories and make it germinating grass and grow it to implement technical problems fed by laws / regulations rain.

It is all about the customer locations and its local laws and regulations, plus Cloud Providers. The location of customer territories is a key factor to determine which legal/technical pair of problems appear according to country, federation or region legislations. CC legal issues drive another two important components which are SLAs/SLOs/QoS. All are related and one can generate others. It is important to explore the legally related technical problems.

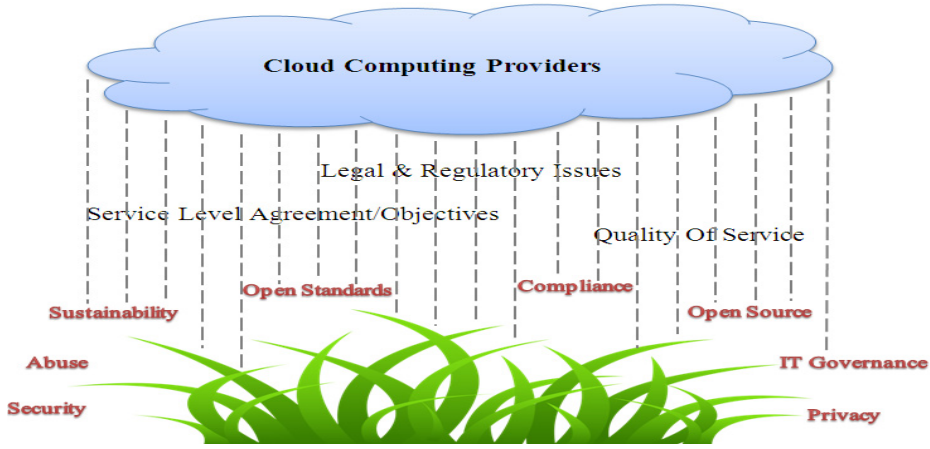


Fig. 2. Technical Problems Inter-Related with Legal Issues

Security, as CC becomes more popular, concerns are being raised regarding the security issues introduced through adoption of this new cloud technology model. The effectiveness and efficiency of traditional protection systems and approaches are being re-evaluated as the characteristics of this innovative deployment model can differ widely from those of traditional architectures [8], [9]. The migration to CC requires an appropriate *IT Governance* model to ensure a secured computing environment and to comply with all relevant organizational information technology policies [10].

Privacy, the CC and its different models have been totally criticized by privacy advocates and experts. Therefore, monitoring of the communication and the data stored between the user and the host company (CP), can be done at will, lawfully or unlawfully [11].

Lack of *Compliance* usually threatens the entire business. In order to obtain company's compliance with laws and regulations such as FISMA, HIPAA, SOX in the United States, and credit card industry's PCI DSS, users may have to move to community or hybrid deployment cloud models [12]. Both are much more expensive and may offer restricted and limited benefits [13].

Open Source, software becomes the foundation for many CPs to implement. This leads to many legal loopholes [14].

Open Standards APIs, are used by CPs. However, they customize these standards to suit their unique implementation and needed specifications. This makes them not interoperable with other CPs [15].

Sustainability, although some CPs are often assumed to be "green computing", there is no published study to proof that. In cold areas where climate favors natural

cooling and renewable electricity is readily available, the environmental effects will be more moderate. Those countries with favorable conditions, such as Finland, Sweden, Norway and Switzerland, are trying to attract CC data centers. Energy efficiency in CC can result from energy-aware scheduling and server consolidation. The more cloud people use, the less energy consumed [16].

Abuse: same as purchasing harmful hardware, customers can purchase the services of CC for nefarious purposes. This includes password cracking and launching attacks using the purchased services and many other attacks depending on the shared equipment used by CPs [17].

2.2 Cloud Computing Legal Problem Components

As we mentioned that CPs may rain to en-grow legally related technical problems, then it is also good to elaborate and address the components of the legal problem which makes the clouds ready to rain those problems on customer's territories.

The below Fig 3, shows the dispute between these three components, in which each component has its own issues which leads the cloud to rain related technical problems.

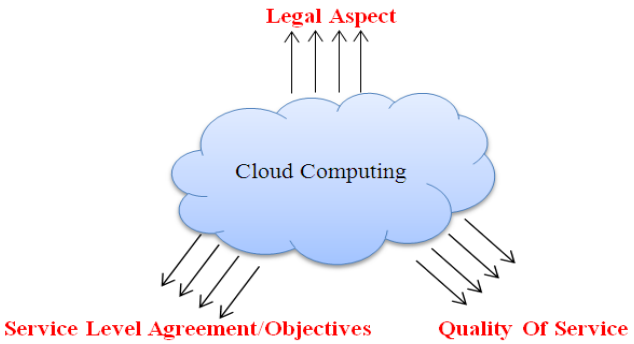


Fig. 3. Legal Problem Components

Legal aspects, are stemming from the lack of understanding of surrounding laws and regulations. In fact legal aspect is an elastic word which carries many meanings according to how the customers of CPs may define it in the light of local and international related legislations.

SLAs/SLOs, companies using CC specifically want their SLAs to be true agreements; negotiated and unique, specific in terms and scope, tied to verifiable facts and unambiguous metrics, unchangeable without notice, unconditioned by externalities, and comparable with SLAs from other vendors. This does not work well with public Cloud SLAs that are just instantiations of global or general terms of service (TOS) statement [18]. There is even more problems, since fine print stipulates the SLA can be changed simply by modifying the TOS where it appears on the company's Website, same as all do [19].

QoS main problem is to maintain and guarantee the quality level agreed upon within the SLO. For example, Privacy and Security are two QoS parameters that present many challenges in cloud-based scenarios [20].

3 Our Proposed Legal Framework

It is a formulated work of analyzing issues, difficulties and challenges appeared at the phase of contracting the CP. We structured our Legal Framework, Fig 4, into two running concurrently phases. Phase1 represents the pyramid includes Legal Aspects, SLA/SLO, QoS processes and their feedback processes (A & B). Phase2 operates continuously using the surrounding cyclic arrows EEEEF (Enforce Enterprise & Environmental Experiences Factors) in the same time with Phase1.

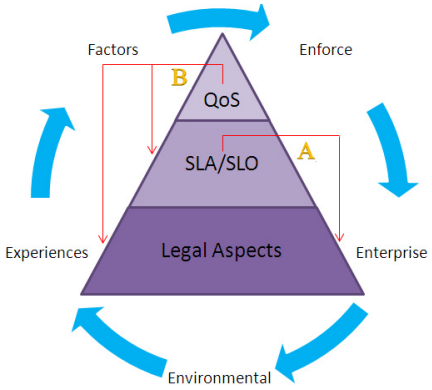


Fig. 4. Proposed Legal Framework Structure

However, companies should operate within these phases together and benefitting from the agility degree offered by the feedback processes (A and B). On the other hand, we assume and consider that a customer decided & ready to move to cloud technology, there are existing CPs for the desired services and the desired services can be outsourced (initially).

3.1 The Main Pyramid

The main pyramid, Fig 5, consists of three processes (Legal Aspects, SLA/SLO & QoS). Each of these processes has its own steps or checks to be visited to guarantee smooth transition to the next process. If any of these steps or checks is missed, then we have the feedback arrows for needed corrective actions.

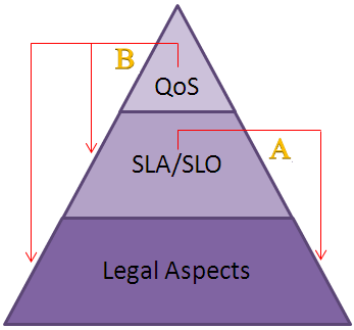


Fig. 5. Phase 1, The Main Pyramid

Process 1: Legal Aspects
At this process, a cloud customer should go through its due diligence. The customer legal department has to determine the legal risk appetite it is ready for. They have to abide to their commitments with other groups, unions, federation, their clients and certificate such as ISOxxxx, SOX or HIPPA. Contracting with CP has to be based on performance evaluation and comparison studies. This has to be based on their reputation, ability to deliver the desired services and many other mandatory evaluating parameters such as financial stability, location, offered infrastructure, prices, capacity, business expandability and elasticity. A broker or negotiator is a value added towards professional SLA/SLO contracts. It is the customer's full responsibility to monitor the contract implementation.

Process 2: SLA/SLO
This process comes as an output from Process1. In fact SLA and SLO are negotiable terms. SLA is a contractual agreement describing the overall service, financial aspects of service delivery, including fees, penalties, bonuses, contract terms and conditions, and specific performance metrics governing compliant service delivery. The metrics of performance indicators are called Service Level Objective (SLOs) [21], [22].

This process interfaces with Process1 in the light of SLA & SLO definitions, and another interfacing with Process3, QoS. The customer has to address the set of SLA/SLO characteristics to be measured such as availability, accessibility, access speed and bandwidth utilization. It is vital that CP has to communicate to his/her customer in advance regarding processing and transferring personal data across the cloud. The feedback process (A) runs continuously to guarantee Process1 and Process2 compatibility.

Process 3: Quality of Service (QoS)
We can call it the pyramid's top, as it is the formulated work which is coming from exercising Process1 and Process2. QoS can be defined as the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance [23].

In this process the metrics have to be measured according to the understanding of services nature. Monitoring QoS does not mean only monitoring SLO metrics, but it should include monitoring any security breaches (like internal or external attacks), unplanned outages, unexpected service interruptions, and other disasters which impact the CP. It is critical to monitor QoS at both CP and customer sides. When monitoring is delegated to the CP, the customer has to have the capability to monitor from his/her side. If the customer is unable to do so, a trusted third-party monitoring company needs to be involved. This is to ensure accurate and non-manipulated monitoring results as well as complex and sophisticated metrics.

As the missing metrics, technical details and specifications may appear during this process, the feedback process (B) takes place to ensure conformity with Process1 and Process2.

3.2 The Surrounding Cyclic Arrows EEEEF

This phase, Fig 6, is developed to be visited during each process of Phase1 processes continuously. The executive managements have to ensure and enforce applying

EEEEF continuously and during Phase1. Moreover it takes place during service design, implementation, transition, operation and termination. EEEEF might be a database, knowledge center or simple documents to be continuously updated with new enterprise and environmental experiences and factors.

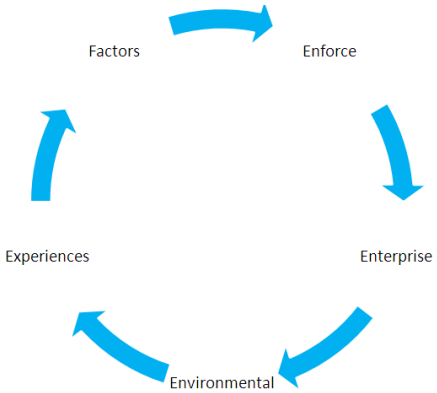


Fig. 6. Phase 2, EEEEF

For example it should contain a list of trusted CPs, specifications, experiences, case studies, real circumstances and situations the customer passed through. Importantly to include risk appetite, Fig 7 which is stemming from laws, regulations and any related updates or changes. The business requirements can be stretched to maximize the benefits without violating the regulatory boundaries.

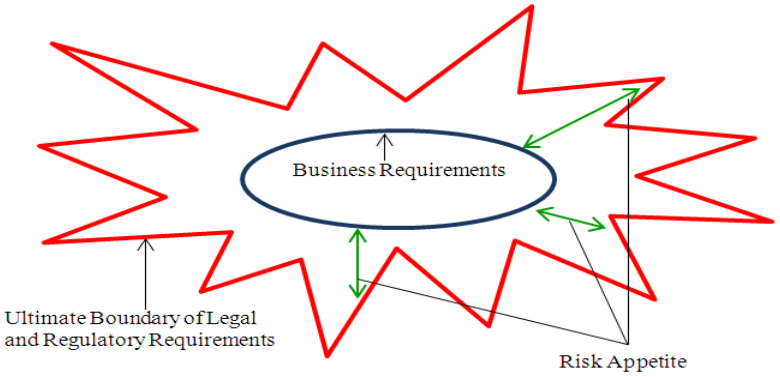


Fig. 7. Risk Appetite

It also should include a list of attributes of Personally Identifiable Information (PII, as shown below in Fig 8), the common known prices for each service category, terms and conditions for standards and certificates compliancy, public periodic technical and non-technical reports, service nature, requirements and technical specifications, lessons learned from previous trading and contracts.

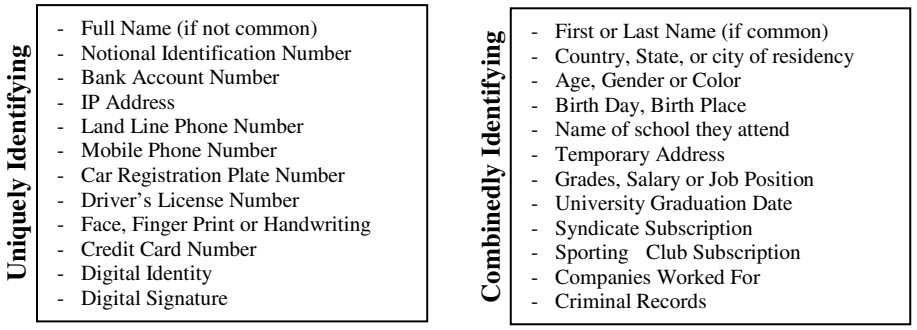


Fig. 8. Personally Identifiable Information (PII)

3.3 IT-Legal Framework Integration Value

Having this IT based Legal framework guarantees smooth and confident transition towards CC. Also, setting clear boundaries and segregations of duties helps to shrink the gray area. However, visiting Legal Aspects, SLA/SLO and QoS processes through Phase1, minimize and control the occurring likelihood of the major technical problems. Thus, the customer territories absorb less rain and germinate to less grass. In addition to that, Legal Framework agility allows the customer to customize it according to new laws and regulations. Our proposed Legal Framework can be more complicated or more simpler according to the environmental and experiences factor as well as organization size.

4 Conclusions and Future Works

Due to lack of standards and ambiguous regulations, the customers seek less IT/Technical pair of risks. Transfer this risk to CP may help, but will generate new IT/Legal pair of risks. In this paper we proposed a novel IT based Legal Framework for CC. It addresses the interlocks between the IT & Legal departments. It is structured into two concurrently running phases. Phase1 assures the compatibility and compliance between the legal problem components. Phase2 accompanies Phase1 to realize the surrounding cyclic arrows EEEEF. Practicing more with our proposed Legal Framework helps the customers to put their hands over areas which need to be developed and enhanced. The customers can add more recent and related processes in Phase1. In addition, involving more and recent factors and experiences in Phase2. The integration with monitoring mechanisms is reserved for future publications.

References

1. Dialogic (2013), http://www.dialogic.com/solutions/cloud-communications/build/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf
2. Chunye, G., Jie, L., Qiang, Z., Haitao, C., Zhenghu, G.: IEEE 39th International Conference on Parallel Processing Workshops, pp. 275–279 (2010)

3. Michael, A., Armando, F., Rean, G., Anthony, D., Randy, K., Andy, K., Gunho, L., David, P., Ariel, R., Ion, S., Matei, Z.: A view of cloud computing. Magazine: Communications of the ACM 53(4), 50–58 (2010)
4. Luis, M., Luis, M., Juan, C., Maik, L.: A Break in the Clouds: Towards a Cloud Definition. Newsletter ACM SIGCOMM Computer Communication Review 39(1), 50–55 (2009)
5. Ian, F., Yong, Z., Ioan, R., Shiyong, L.: Cloud Computing and Grid Computing 360-Degree Compared. In: IEEE Grid Computing Environments Workshop (2009)
6. Search Cloud Security (2013), http://searchcloudsecurity.techtarget.com/tutorial/Cloud-computing-legal-issues-Developing-cloud-computing-contracts
7. Chazalet, A.: Service Level Checking in the Cloud Computing Context. In: IEEE 3rd International Conference on Cloud Computing, pp. 297–304 (2010)
8. Subashini, S., Kavitha, V.: A survey on Security Issues in Service Delivery Models of Cloud Computing. Journal of Network and Computer Applications 34(1), 1–11 (2011)
9. Brian, H., Kara, N., Matt, B.: Storm Clouds Rising: Security Challenges for IaaS Cloud Computing. In: IEEE 44th Hawaii International Conference on System Sciences (HICSS), pp. 1–7 (2011)
10. Manish, P., Jong, P.: Cloud computing: future solution for e-governance. In: ACM Proceedings of the 3rd International Conference on Theory and Practice of Electronic Governance, pp. 409–410 (2009)
11. Hassan, T., James, B., Gail-Joon, A.: Security and Privacy Challenges in Cloud Computing Environments. IEEE Security & Privacy Magazines 8(6), 24–31 (2010)
12. Spyridon, V., Gregory, Y.: Wireless Going in the Cloud: A Promising Concept or Just Marketing Hype? Journal of Wireless Personal Communications 58(1), 5–16 (2011)
13. Dave, D.: Why Cloud Computing Will Never Be Free. Magazine: Queue – Emulators ACM 8(4), 62–69 (2010)
14. Daniel, N., Rich, W., Chris, G.: The Eucalyptus Open-source Cloud-computing System. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 124–131 (2009)
15. Alex, G., Kostantinos, S., Michael, Z.: Efficient Deployment of Predictive Analytics through Open. ACM SIGKDD Explorations Newsletter 11(1), 32–38 (2009)
16. Marinos, A., Briscoe, G.: Community Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 472–484. Springer, Heidelberg (2009)
17. Siani, P., Azzedine, B.: Privacy, Security and Trust Issues Arising from Cloud Computing. In: IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 693–702 (2010)
18. Smarter Technology (2013), http://www.smartertechnology.com/c/a/Cloud-Computing/Problems-With-SLAs-for-Cloud-Services/
19. Wang, L., Von Laszewski, G., Kunze, M., Tao, J.: Cloud Computing: a Perspective Study. Journal of New Generation Computing 28(2), 137–146 (2010)
20. Qi, Z., Lu, C., Raouf, B.: Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1(1), 7–18 (2010)
21. Search Cloud Security (2013), http://expertanswercenter.techtarget.com/eac/knowledgebaseAnswer/0,295199,sid63_gci1175490,00.html
22. Rabi, P., Manas, P., Suresh, S.: SLAs in Cloud Systems: The Business Perspective. International Journal of Computer Science and Technology 3(1), 481–488 (2012)
23. Paul, H., Shrisha, R., Charles, S., Akshay, N.: System of Systems to Provide Quality of Service Monitoring, Management and Response in Cloud Computing Environments Raytheon Intelligence and Information Systems, pp. 1–17 (2012)

A Blind Robust 3D-Watermarking Scheme Based on Progressive Mesh and Self Organization Maps

Mona M. Soliman[1,2], Aboul Ella Hassanien[1,2], and Hoda M. Onsi[1]

[1] Cairo University, Faculty of Computers and Information, Cairo, Egypt
[2] Scientific Research Group in Egypt (SRGE)
http://www.egyptscience.net

Abstract. Most of progressive mesh (PM) transmission techniques, consist in iteratively decimating the mesh, while storing the information necessary to the process inversion. During the transmission or visualization the 3D content can be duplicated and redistributed by a pirate. Digital watermarking is considered as a good solution to this emerging problem. This paper focus on introducing a novel robust and blind mesh watermarking schema by converting the original triangle mesh into a multi-resolution format, consisting of a coarse base mesh and a sequence of refinement, the watermark bits are inserted through progressive mesh level of details, and extracted at refinement stage without any need for the original model. The watermark insertion is performed only on set of marked vertices come out from Self Organization Maps (SOM) clustering neural network. These vertices are used as candidates for watermark carriers that will hold watermark bits through progressive mesh transmission. The robustness of proposed techniques is evaluated experimentally by simulating attacks such as mesh smoothing, noise addition and mesh cropping.

1 Introduction

The 3D geometric model is one of the most fundamental techniques in the fields of computer-aided design (CAD), computer-assisted manufacturing (CAM), computer-aided engineering (CAE), and computer graphic (CG) animation [1]. The current situation of the distributed engineering environment increases opportunities to globally exchange digital data of geometric models between various organizations through a computer network. This also increases the possibility of theft of the geometric model by illegal duplication. Therefore the copyright property of the geometric model must be strongly protected from theft [2].

Since 3-D mesh watermarking techniques were introduced, there have been several attempts to improve the performance in terms of transparency and robustness. Robustness is achieved when the watermark can be retrieved even after the watermarked model has been processed or attacked intentionally specific algorithms. Depending on whether the original cover-media is needed or not in the detection stage we have non-blind and blind watermarking. The methods

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 131–142, 2013.

from the first category usually have good robustness, but they are not suitable for most applications [5].

There are two kinds of 3D mesh watermarking algorithms that are similar to image watermarking. One is based on the spatial domain [3]- [6], which provides many insights into mesh watermarking. [7]- [10] suggest several spatial domain based algorithms in which the blindness has been achieved. These schemes has a certain level of robustness against common geometry attacks and even cropping, but are in general fragile to connectivity attacks because the used geometric watermarking primitives may disappear or be seriously disturbed after such attacks. The other algorithms are based on the transformation domain [11]- [17]. In these algorithms, spectral decomposition and multi resolution techniques such as wavelet transform and progressive meshes are used to decompose a 3D model into a lower resolution and the watermark is inserted in the bit stream. The model is then reconstructed from lower resolutions.

Progressive meshes is one of the multi resolution techniques. This technique was introduced by Hoppe in [18]. Progressive mesh introduced as a new format for storing and transmitting arbitrary triangle meshes. For a given mesh, the PM representation defines a continuous sequence of level-of-detail (LoD) approximations, allows smooth visual transitions (geomorphs) between these approximations, supports progressive transmission, and makes an effective compression scheme. In short, progressive meshes offer an efficient, lossless, continuous-resolution representation.

Progressive compression [20] allows to achieve high compression ratio (and thus fast transmission) and also to produce different levels of detail (LoD), allowing to adapt the complexity of the data to the remote device by stopping the transmission when a sufficient LoD is reached. Most of progressive compression techniques, consist in iteratively decimating the mesh (vertex/edge suppressions), while storing the information necessary to the process inversion, i.e. the refinement. The problem facing the progressive mesh compression that the transmitted model can be duplicated and redistributed by a pirate, so there is a need to protect such transmission from attacks and protect it from illegal duplication.

This paper focus on introducing a robust and blind mesh watermarking schema by converting the original triangle mesh into a multi-resolution format, consisting of a coarse base mesh and a sequence of refinement, the watermark bits are inserted through progressive mesh level of details, and extracted at refinement stage without any need for the original model. Watermark insertion is performed on specific set of vertices that are selected by utilizing Self Organization Maps (SOM) [19]. SOM is a kind of competitive neural network in which the networks learn to form their own classifications of the training data without external help. Watermark bits sequence are inserted based on the mean and stander deviation of selected vertices' neighbours in the original mesh.

The remainder of this paper is organized as follows. Section (2) reviews related work of watermark 3D model within progressive mesh. Section (3) illustrate some useful and important preliminary ideas relating to this work. Section (4)

discusses the proposed watermarking scheme using both SOM and PM . Section (5) shows the experimental results. Conclusions are discussed in Section (6).

2 Related Work

The concept of progressive mesh compression was introduced for the first time by Hoppe in [18]; by generating lower resolution models of the mesh by performing a sequence of vertex split operations. A set of methods were then introduced, [21]-[22], to improve the compression ratio by applying the collapse/split operations on sets of independent vertices. Other methods are based on vertex removal, they consist in removing sequences of vertices and re-triangulating the holes left by the deletions at no cost [20]. In [23] Peng aims at carefully preserving the relevant features at each intermediate level of detail using a hierarchical clustering. In [24], the authors proposed a new progressive approach based on a reconstruction scheme, the algorithm starts from a coarse version of the original model which is refined progressively by inserting a vertex to the longest edge using edge split operation, aiming to generate uniformly sampled intermediate meshes.

Having improvement in the techniques of the progressive compression lead into the question how to protect this content during its transmission. Many attemps try to answer this question by utilizing 3D watermark procedures. Praun and Hoppe [12] reported robust mesh-watermarking algorithm that works using Hoppe's progressive meshes. Such algorithm provide a scheme for constructing a set of scalar basis functions over the mesh vertices. Then they adapt the spread-spectrum principles used in image watermarking to embed information into the basis functions corresponding to perceptually significant features of the model. Chen and Chen [25] proposed 3D mesh watermarking approach on the basis of the progressive mesh and the discrete wavelet transform. By embedding the transformed sequence of the watermark image into the vertex split sequence of the progressive mesh, the watermark and the cover object are transmitted synchronously such that the progressive decode of the cover object and the extraction and decoding of the watermark can be performed on-the-fly with the transmission. Lee [20] presented a joint reversible watermarking and progressive compression of 3D meshes. Each LoD is compressed and watermarked by modifying the geometry of refined vertices with respect to the center of mass of the original 3D mesh. The watermark process is reversible in the sense that the geometrical modifications introduced by the embedding processing can be removed after watermark extraction.

3 Preliminaries

3.1 Self Organizing Maps

Self-Organizing means no supervision is required. SOMs learn on their own through unsupervised competitive learning. Maps means they attempt to map

their weights to conform to the given input data [26]. In competitive learning, the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time. This activated neuron is called a winner-takes-all neuron or simply the winning neuron. Such competition can be induced/implemented by having lateral inhibition connections (negative feedback paths) between the neurons. The result is that the neurons are forced to organize themselves [27], the n-dimensional input is processed by exactly the same number of computing units as there are clusters to be individually identified as shown in Figure 1. In this work we utilize the particular kind of SOM known as a Kohonen Network. This SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer. Clearly, a one dimensional map will just have a single row (or a single column) in the computational layer.

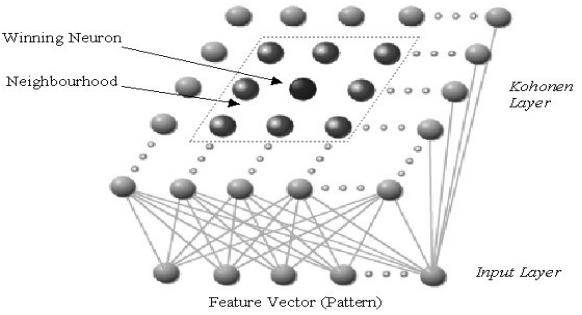


Fig. 1. Structure of SOM Kohonen network [29]

3.2 Progressive Mesh

In the progressive mesh (PM) representation as introduced by Hoppe [18], an arbitrary mesh $\hat{M}$ is stored as a much coarser mesh M^0 together with a sequence of n detail records that indicate how to incrementally refine M^0 exactly back into the original mesh. Each of these records stores the information associated with a vertex split, an elementary mesh transformation that adds an additional vertex to the mesh. The PM representation of $\hat{M}$ thus defines a continuous sequence of meshes M^0,M^1,...,M^n of increasing accuracy, from which LOD approximations of any desired complexity can be efficiently retrieved. Edge collapse, is an operation that is sufficient for effectively simplifying meshes. As shown in Figure 2, an edge collapse transformation unifies 2 adjacent vertices v_1 and v_2 into a single vertex v_1. The vertex v_2 and the two adjacent faces f_1 and f_2 vanish in the process. A position v_1 is specified for the new unified vertex. A key observation is that an edge collapse transformation is invertible. [18] call that inverse transformation a vertex split. A vertex split transformation adds near vertex v_1 a new vertex v_2 , and two new faces f_1 and f_2.

Because edge collapse transformations are invertible, it can represent an arbitrary triangle mesh $\hat{M}$ as a simple mesh M^0 together with a sequence of n v_{split} records. Hoppe call $(M^0, (v_{split_0}, v_{split_1}, \ldots, v_{split_{n-1}}))$ a progressive mesh (PM) representation of $\hat{M}$.

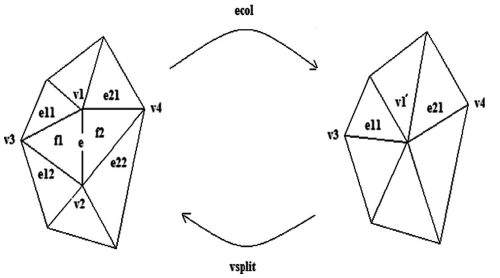


Fig. 2. Edge collapse operation

4 The Proposed 3D Watermarking Scheme

Most of progressive compression techniques, consist in iteratively decimating the mesh, while storing the information necessary to the process inversion, i.e. the refinement [20]. During the transmission or visualization the 3D content can be duplicated and redistributed by a pirate. Digital watermarking is considered as a good solution to this emerging problem. The main problem is how to combine compression and watermark sequence. A simple solution consists in inserting the watermark bit sequence during the decimation process. since the mesh is transmitted in a progressive way until a certain level of detail (not necessary the finest one), the watermark have to be readable in all intermediate LoDs. The watermark insertion is performed only on set of marked vertices come out from SOM neural network as candidates for watermark carriers. Watermark bit sequence are inserted based on the mean and standard deviation of marked vertices' neighbors in the original mesh. The Extraction is performed during the refinement sequence without any need to the original model. Figure 3 show the whole watermarking scheme for both insertion and extraction. The following subsection illustrate in more details the basic steps necessary of performing watermark insertion and extraction.

4.1 Vertex Selection Using SOM

The first step in our proposed scheme is to select set of vertices and mark it as watermark carrier. We aim to make this selection in an intelligent way that guarantee preserving model from distortion during watermark insertion process. Based on this requirement we utilize the SOM neural network in clustering the vertices of original mesh into suitable and non-suitable candidates for watermark carriers. This can be performed as proposed in our previous work [19] by clustering vertices based on smoothness feature. The smoothness feature measure the

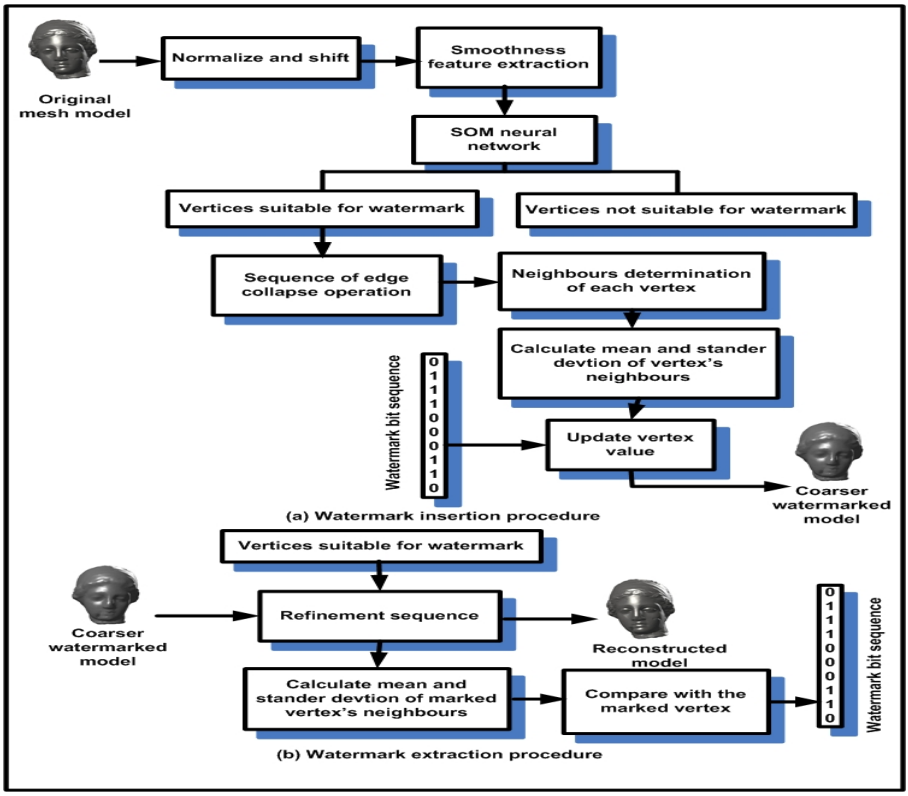


Fig. 3. The proposed watermarking scheme

angle variation between surface normals and the average normal corresponding to a vertex. The feature vector extraction method results in a set of angles derived by computing the orientation of the surface normals to the average normal of the triangular faces that form a 1-ring neighbourhood for a vertex. Such a smoothness measure reflect the local geometry of a surface or region. Flat and peak surface are not suitable for inserting watermark bits sequence.

4.2 Watermark Insertion Procedure

The watermark insertion is performed with each edge collapse iteration. Since each edge collapse result in one vertex removal and two faces removal, the collapse iteration is performed around marked vertices resulting from SOM clustering. The selected edge for collapse contain a vertex come out from the SOM clustering and this vertex is used to carry the watermark bit, The other vertex is eliminated. The insertion of watermark bits sequence w is performed by searching for the neighbors of marked vertices, estimating their mean and standard deviation and insert watermark using these two values.

Basic steps of watermark insertion procedure is illustrated in algorithm 1

Algorithm 1. The watermark insertion procedure

Input: list of vertex positions, list of faces, marked vertex as watermark carriers, watermark bit sequence w

for Each iteration of edge collapse **do**

Call the marked vertex as watermark carrier.

Create a list of marked vertex's immediate neighbors $V\{NH\}$.

Determine the edges connecting marked vertex and its neighbors.

Pick the first neighbor $V\{NH_1\}$

$\Rightarrow$ Eliminate $V\{NH_1\}$ from the neighbors list.

$\Rightarrow$ Eliminate $V\{NH_1\}$ from the vertex mesh model.

$\Rightarrow$ Eliminate two faces connecting $V\{NH_1\}$ with marked vertex from the face list.

Calculate mean and standard deviation of marked vertex neighbors' list.

Update the value of marked vertex according to the following rule.

if watermark bit w_i=1 **then**

$$\begin{aligned} V_x^{'} &= \mu(V\{NH_x\}) + 2 * \sigma(V\{NH_x\}) \\ V_y^{'} &= \mu(V\{NH_y\}) + 2 * \sigma(V\{NH_y\}) \\ V_z^{'} &= \mu(V\{NH_z\}) + 2 * \sigma(V\{NH_z\}) \end{aligned} \tag{1}$$

end if

if watermark bit wi==0 **then**

$$\begin{aligned} V_x^{'} &= \mu(V\{NH_x\}) - 2 * \sigma(V\{NH_x\}) \\ V_y^{'} &= \mu(V\{NH_y\}) - 2 * \sigma(V\{NH_y\}) \\ V_z^{'} &= \mu(V\{NH_z\}) - 2 * \sigma(V\{NH_z\}) \end{aligned} \tag{2}$$

end if

end for

Output: sequence of coarser watermarked mesh model

4.3 Watermark Extraction Method and 3D Mesh Reconstruction

Here we utilize a blind watermarking procedure thus there is no need to the original model during the extraction process. To extract the watermark, we need the position of marked vertices as watermark carrier used in the insertion procedure. Therefore, we need to store these data for the correct watermark extraction. Once these locations are detected watermark bits w are extracted by comparing the value of marked vertices with their immediate neighbors' mean and standard deviation.This comparison is performed by a shifted value α to take consideration of high correlation in case attacks is happen. During the Extraction procedure the current intermediate mesh is refined with insertion of a set of vertices to formulate the collapsed edge around marked vertex carried the watermark bit. Algorithm 2 illustrate in more details the basic steps of watermark extraction procedure.

Algorithm 2. The watermark extraction procedure

Input: sequence of coarser watermarked mesh model,marked vertex as watermark carriers.
for Each iteration of model refinement. **do**
Call the marked vertex as watermark carrier.
Create a list of marked vertex's immediate neighbors $V\{NH\}$
$\Rightarrow$ Predict the vertex $V\{NH_1\}$ that eliminated during edge collapse iteration according to its neighbor.
$\Rightarrow$ Update coarser mesh model with the predicted vertex.
$\Rightarrow$ Update the face list of the coarser model.
Calculate mean and standard deviation of marked vertex neighbors' list.
Compare the marked vertex with calculated mean and standard deviation.

$$\begin{array}{l} \mu_1 x = \mu(V\{NH_x\}) + 2 * \sigma(V\{NH_x\}) \; \mu_2 x = \mu(V\{NH_x\}) - 2 * \sigma(V\{NH_x\}) \\ \mu_1 y = \mu(V\{NH_y\}) + 2 * \sigma(V\{NH_y\}) \; \mu_2 y = \mu(V\{NH_y\}) - 2 * \sigma(V\{NH_y\}) \\ \mu_1 z = \mu(V\{NH_z\}) + 2 * \sigma(V\{NH_z\}) \; \mu_2 z = \mu(V\{NH_z\}) - 2 * \sigma(V\{NH_z\}) \end{array} \tag{3}$$

if $(V_x^{'} \geq \mu_1 x\text{-}\alpha) \& (V_x^{'} \leq \mu_x + \alpha)$ **then**

$$w_i = 1; \tag{4}$$

end if
if $(V_x^{'} \geq \mu_2 x\text{-}\alpha) \& (V_x^{'} \leq \mu_x + \alpha)$ **then**

$$w_i = 0; \tag{5}$$

end if
end for
Output:Reconstructed model, watermark bit sequence w

5 Experimental Results and Discussion

The proposed algorithm is developed in $MATLAB7.6$ environment. Several models are used for the evaluation and three of them are shown in Figure 4 : bunny (34,835 vertices;69,666 faces), dragon (50,000 vertices; 100,000 faces), venus (100,579 vertices;201,514 faces), The watermark bits are generated randomly by a length of 50 bits. This work is compared to the algorithm of Cho et al. [28] that considered one of the best 3D watermarking algorithm in terms of robustness and blindness. Cho in [18] proposed n watermark insertion in the vertex norm distribution.

5.1 Distortion Evaluation

Once edge collapse is performed the mesh model is stored as coarser base mesh and a sequence of refinements. Reconstruction of the original mesh from the

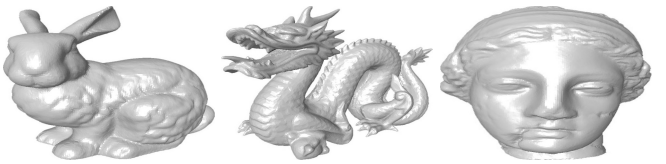


Fig. 4. Original 3D mesh models

coarser base mesh introduce a smoothed model close to the original one but not identical to it. Since the watermark is inserted during the edge collapse iterations the reconstructed model is degraded due to the presence of watermark bits sequence. Vertex Signal-to-Noise Ratio (VSNR) quantify the visual differences between the original and watermarked models. In this work it also measure the distortion between the original model and the reconstructed model holding watermark bit sequence. Table 1 shown the VSNR values for the tested models using different refinement iterations. Obliviously as number of refinement iterations increased, more smoothed model is generated, and the VSNR is decreased but as shown in the Table 1 this is performed by very small rate.

Table 1. VPSNR values at different refinement levels M

Model	$(M = 50)$	$(M = 80)$	$(M = 120)$
Bunny	104.40	104.19	103.98
Dragon	113.95	113.17	112.19
Venus	137.16	136.24	135.14

5.2 Robustness Evaluation

To investigate the robustness of watermark scheme, each watermarked model is attacked by simulating set of geometric attacks such as: • Additive binary noise : random noise is added to the watermarked model with different error rates (e.g. Amp= 0.005, 0.009, 0.01, 0.03, 0.05). Here, the error rate represents the noise amplitude as a fraction of the maximum vertex norm of the object.

• Smoothing attack: Laplacian smoothing is applied to the watermarked model smooths the sharp edges in the model by applying a low pass gradient filter to the vertices. (e.g. setting λ=0.03 with different iterations N_{it}=5, 10, 15,20)

• Cropping attack : cropping is performed by cutting 3D model with different levels indicating the percentage of cropped vertices (e.g. 3%,5%,10%,20%).

We give here the robustness evaluation against only common geometric attacks. As our algorithm is based on a connectivity-based compression technique, the refinement is not possible when a change of connectivity occurs. Hence our

algorithm is not robust to connectivity change. This work is compared with approach in [28] for both random noise attacks and smoothing noise attacks, clipping attack simulation shows that Cho et. al in [18] approach are very vulnerable to such attacks that cause severe alteration to the center of gravity of the model so it does not shown in the reported results.

The robustness is evaluated in terms of correlation coefficient between the extracted watermark bit sequence $w_n^{'}$ and the originally inserted one w_n as given by the following equation [19]:

$$Corr = \frac{\sum_{n=1}^{N-1}(w_n^{'} - w^{-'})(w_n - w^{-})}{\sqrt{\sum_{n=1}^{N-1}(w_n' - w^{-'})^2 \sum_{n=1}^{N-1}(w_n - w^{-})^2}} \tag{6}$$

Where $w^{-'}$ and w^{-} indicate respectively the averages of the watermark bit sequence $w_n^{'}$ and w_n. This correlation value measures the similarity between two strings and varies between 1 (orthogonal sequence) and +1 (the same sequence). Tables 2-4 show the correlation results for different types of attacks with different parameters.

The proposed approach as shown in Tables (2,3, and 4) is superior that Cho approach in terms of geometric attacks, the proposed method can not cope with connectivity attacks.

Table 2. Evaluation of Robustness Against Additive Noise Attacks

	Proposed Scheme			Cho Approach		
Amp	Bunny	Dragon	Venus	Bunny	Dragon	Venus
0.005	1	0.95	0.90	0.28	0.46	0.41
0.009	0.90	0.75	0.54	0.17	0.22	0.15
0.01	0.83	0.56	0.66	0.16	0.20	0.06
0.03	0.43	0.31	0.20	0.14	0.21	0.19
0.05	0.28	0.28	0.08	0.12	0.22	0.15

Table 3. Evaluation of Robustness Against Smoothing Attack ($\lambda = 0.03$)

	Proposed Scheme			Cho Approach		
N_{it}	Bunny	Dragon	Venus	Bunny	Dragon	Venus
5	1	0.954	0.954	1	1	0.95
10	0.95	0.80	0.95	0.90	0.95	0.84
15	0.66	0.546	0.80	0.90	0.83	0.83
20	0.61	0.54	0.75	0.85	80	0.87

Table 4. Evaluation of Robustness Against Cropping Attack (Cropping %)

%	Bunny	Dragon	Venus
3	0.90	0.85	0.80
5	0.90	0.75	0.65
10	0.71	0.546	0.45
20	0.71	0.36	0.45

6 Conclusions and Future Work

In this paper, we have presented a novel approach of combining watermarking within progressive compression. Each LoD is compressed and watermarked by modifying a marked vertex selected by SOM as suitable watermark carrier. This modification of vertex value is performed with respect to the mean and stander deviation of its immediate neighbours in the original 3D mesh. The proposed method is robust against different geometric attacks like random noise, smoothing,and even cropping. For future works, we plan to extend the robustness of our method to connectivity attacks.

References

1. Shinichi, M., Y.: Watermarking for 3D Polygons Using Wavelet Transform and Modified Traveling Salesman Problem. Journal of the Operations Research Society of Japan 52(4), 402–416 (2009)
2. Kanai, S., Data, H., Kishinami, T.: Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition. In: Proceedings of 6th IFIP WG 5.2 GEO-6, pp. 296–307 (1998)
3. Ohbuchi, R., Aono, M.: Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications. IEEE Journal on Selected Areas in Communications 16, 551–560 (1998)
4. Benedens, O.: Geometry-based watermarking of 3D models. IEEE Computers and Applications 19(1), 46–55 (1999)
5. Yeo, B., Yeung, M.M.: Watermarking 3D objects for verification. IEEE Computers and Applications 19(1), 36–45 (1999)
6. Ohbuchi, R., Masuda, H., Aono, M.: Watermarking Three Dimensional Polygonal Models Through Geometric and Topological Modifications. IEEE Journal on Selected Areas in Communications 16(4), 551–560 (1998)
7. Ohbuchi, R., Masuda, H., Aono, M.: Data Embedding Algorithms for Geometrical and non-Geometrical Targets in Three-Dimensional Polygonal Models. Computer Communications 21(15), 1344–1354 (1998)
8. Cayre, F., Macq, B.: Data Hiding on 3-D Triangle Meshes. IEEE Trans. on Signal Processing 51(4), 939–949 (2003)
9. Agarwal, P., Prabhakaran, B.: Robust Blind Watermarking Mechanism for Point Sampled Geometry. In: Proc. of the ACM Multimedia and Security Workshop, pp. 175–186 (2007)

10. Wang, Y.P., Hu, M.: A New Watermarking Method for 3D Models Based on Integral Invariants. IEEE Trans. on Visualization and Computer Graphics 15(2), 285–294 (2009)
11. Valette, S., Prost, R.: Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes. IEEE Trans. on Visualization and Computer Graphics 10(2) (2004)
12. Praun, E., Hoppe, H., Finkelstein, A.: Robust Mesh Watermarking. In: SIGGRAPH Proceedings, pp. 69–76 (1999)
13. Kanai, S., Date, H., Kishinami, T.: Digital Watermarking for 3D Polygons Using Multi-Resolution Wavelet Decomposition. In: Proceedings of the Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6), Japan, pp. 296–307 (1998)
14. Yin, K.K., Pan, Z.G., Shi, J.Y., Zhang, D.: Robust Mesh Watermarking Based on Multiresolution Processing. Computers & Graphics 25(3), 409–420 (2001)
15. Guskov, I., Sweldens, W., Shroder, P.: Multiresolution Signal Processing for Meshes. In: Proceedings of SIGGRAPH 1999, pp. 49–56 (1999)
16. Ohbuchi, R., Takahashi, S., Miyazawa, T., Mukaiyama, A.: Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain. In: Proceedings of the Graphics Interface, Canada, pp. 9–17 (2001)
17. Ohbuchi, R., Takahashi, S.: A frequency Domain Approach to Watermarking 3D Shapes. In: EUROGRAPHICS, Saarbrucken, Germany, vol. 21(3), pp. 2–6 (2002)
18. Hoppe, H.: Progressive Mesh. ACM SIGGRAPH 96, 99–108 (1996)
19. Soliman, M.M., Ella Hassanien, A., Onsi, H.M.: Robust Watermarking Approach for 3D Triangular Mesh using Self Organization Map. Submitted in Federated Confrence on Computer Science and Information System, Krakw, Poland (September 2013)
20. Lee, H., Dikici, C., Lavoué, G., Dupont, F.: Joint Reversible Watermarking and Progressive Compression of 3D Meshes. Visual Computer 27(6-8), 781–792 (2011)
21. Pajarola, R., Rossignac, J.: Compressed Progressive Meshes. IEEE Transactions on Visualization and Computer Graphics 6(1), 79–93 (2000)
22. Taubin, G., Gueziec, A., Horn, W., Lazarus, F.: Progressive Forest Split Compression. In: ACM SIGGRAPH, pp. 123–132 (1998)
23. Peng, J., Kuo, Y., Eckstein, I., Gopi, M.: Feature Oriented Progressive Lossless Mesh Coding. Computer Graphics Forum 29(7), 2029–2038 (2010)
24. Valette, S., Chaine, R., Prost, R.: Progressive Lossless Mesh Compression via Incremental Parametric Refinement. In: Proceedings of the Symposium on Geometry Processing, vol. 28, pp. 1301–1310 (2009)
25. Hung-Kuang, C., Yung-Hung, C.: Progressive Watermarking on 3D Meshes. In: IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1–7 (2010)
26. Guthikonda, S.M.: Kohonen Self-Organizing Maps (2005)
27. Rojas, R.: Neural Networks A Systematic Introduction, A book Foreword by Jerome Feldman (502 p. 350 illustrations). Springer, Berlin (1996)
28. Cho, J.W., Prost, R., Jung, H.Y.: An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms. IEEE Transaction on Signal Processing 55(1), 144–152 (2005), doi:10.1007/11551492-24
29. Richard, M.C.: An incremental learning system based on features derived using fast Gabor transforms for the identification of textural objects. In: Proceedings of SPIE, Vision Geometry X, International Symposium on Optical Science and Technology, San Diego, California, USA, vol. 4476, pp. 109–119 (2001)

A Cattle Identification Approach Using Live Captured Muzzle Print Images

Ali Ismail Awad[1,⋆], Aboul Ella Hassanien[2,⋆⋆], and Hossam M. Zawbaa[3,⋆]

[1] Faculty of Engineering
Al Azhar University, Qena, Egypt
[2] Faculty of Computers & Information
Cairo University, Cairo, Egypt
[3] Faculty of Computers and Information
BeniSuef University
BeniSuef, Egypt
aawad@ieee.org, {aboitcairo,hossam.zawbaa}@gmail.com

Abstract. Cattle identification receives a great research attention as a dominant way to maintain the livestock. The identification accuracy and the processing time are two key challenges of any cattle identification methodology. This paper presents a robust and fast cattle identification approach from live captured muzzle print images with local invariant features. The presented approach compensates some weakness of traditional cattle identification schemes in terms of accuracy and processing time. The proposed scheme uses Scale Invariant Feature Transform (SIFT) for detecting the interesting points for image matching. In order to enhance the robustness of the presented technique, a Random Sample Consensus (RANSAC) algorithm has been coupled with the SIFT output to remove the outlier points and achieve more robustness. The experimental evaluations prove the superiority of the presented approach because it achieves 93.3% identification accuracy in reasonable processing time compared to 90% identification accuracy achieved by some other reported approaches.

1 Introduction

Recently, governments pay a great attention to the livestock by providing vaccination to the most of the diseases. They seek to overcome some food problems and keep the livestock as huge as possible. Cattle identification plays an important role in controlling the disease outbreak, vaccination management, production management, cattle traceability, and assigning ownership [1]. Traditional cattle identification methods such as ear notching, tattooing, branding, or even some electrical identification methods, such as the Radio Frequency Identification (RFID) [2], are not able to provide any reliable cattle identification due to theft, fraudulent and duplication. Therefore, the need to a robust identification scheme is a vital must. Although, the identification and recognition modes are valid for cattle animals, this research focuses on the cattle identification mode.

⋆ Member of the Scientific Research Group in Egypt (SRGE).
⋆⋆ Chairman of the Scientific Research Group in Egypt (SRGE).

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 143–152, 2013.

Human biometrics is a key fundamental security mechanism that assigns unique identity to an individual according to some physiological or behavioral features [3], [4]. These features are sometimes named as biometrics modalities, identifiers, traits or characteristics. Human biometrics identifiers must fulfill some operational and behavioral characteristics such as uniqueness, universality, acceptability, circumvention and accuracy [5], [6]. Adopting human biometrics identifiers into animals is a promising technology for cattle identification domain, and it has many applications such as cattle classification, cattle tracking from birth to the end of food chain, and understanding animal diseases' trajectory. On the other side, using animal biometrics in computerized systems faces great challenges with respect to accuracy and acceptability as the animal movement can not be easily controlled. Thus, adopting human biometrics to animals may solve plenty of identification challenges.

Muzzle print, or nose print, was investigated as distinguished pattern for animals since 1921 [7]. It is considered as a unique animal identifer that is similar to human fingerprints. Paper-based or inked muzzle print collection is inconvenient and time inefficient process. It needs special skill to control the animal and get the pattern on a paper. Furthermore, the inked muzzle print images do not have sufficient quality, and can not be used in a computerized manner [8]. Thus, there is a lack of a standard muzzle print benchmark. Driven from this need, the first contribution of this research is to collect a database of live captured muzzle print images that works as a benchmark for the proposed cattle identification approach. The standardization of the muzzle prints database is a future need.

A local feature of an image is usually associated with a change of an image property such as texture, color, and pixel intensity [9]. The advantage of local features is that they are computed at multiple points in the image, and hence they are invariant to image scale and rotation. In addition, they do not need image pre-processing or segmentation [10]. Scale Invariant Feature Transform (SIFT) [9], [11] is one of the popular methods for image matching and object recognition. SIFT has been used by some researchers in human biometrics with applications on fingerprints [12], [13], [14] and palmprints [15]. It efficiently extracts robust features; therefore, it has been used to overcome different image degradations such as image noise, partiality, scale, shift, and rotation.

The identification accuracy is the foremost important factor for measuring the performance of any cattle identification approach. This paper presents a robust cattle identification approach that uses a SIFT features for calculating the similarity score between the input muzzle print image and the template one. The superiority of the proposed technique is the assured cattle identification robustness provided by combining the robust SIFT features with a RANdom SAmple Consensus (RANSAC) for robust SIFT features matching [16].

The reminder part of this paper is organized as follows: Section 2 represents the human biometrics technology, and the qualification criteria for selecting biometrics identifiers. Section 3 explains the architecture of the proposed approach, and the implementation phase. Section 4 explores the evaluation phase of the proposed approach. Conclusions and future work are reported in Section 5.

Table 1. Comparison between different biometrics identifiers: 1 = High, 0.5 = Medium and 0 = Low*

	Universality	Uniqueness	Performance	Acceptability	Circumvention	Score
Fingerprint	0.5	1.0	1.0	0.5	1.0	**4.0**
Face image	1.0	0.0	0.5	1.0	0.0	2.5
Iris pattern	1.0	1.0	1.0	0.0	1.0	**4.0**
DNA	1.0	1.0	1.0	0.0	0.0	**3.0**
EEG	1.0	0.0	0.0	0.0	0.0	1.0
Signature	0.0	0.0	0.0	1.0	0.0	1.0
Voice	0.5	0.0	0.0	1.0	0.0	1.5
Gait	0.5	0.0	0.0	1.0	0.0	1.5

*The table is adopted from [17], [18]

2 Human Biometrics Technology

Biometrics modalities provide a high security level with preserved accuracy and reliability for its automated authentication or identification systems. Biometric authentication compensates some weaknesses of token- and knowledge-based traditional authentication approaches by replacing "something you poisson" or "something you know" by "something you are" [19], [20]. It offers not only an automatic authentication method, but also a convenience to the user, not having to remember information or carry a poisson [21], [22]. Driven from its merits, biometrics technology deployment is kept disseminating with large industrial revenue and investments, and it is ongoing fundamental technology for future personal, mobile and governmental applications, [21], [23].

The enormous needs of biometrics deployments in civilian or forensic applications, a large number of biometrics traits have been discovered by taking advantages of the enhanced understanding of the human body [24]. A qualified biometrics trait must be investigated and filtered through the selection criteria. The candidate biometrics identifier should achieve some technical and operational requirements according to the type of application. The competency requirements might be summarized as [17], [25]:

- **Universality**, in terms that the selected identifier must be available for each individual, and the identifier can be measured quantitatively without affecting the user privacy or health.
- **Uniqueness**, which indicates that the selected identifier should contain enough features to differentiate between two persons carrying the same trait. The identifier should be time invariant.

- **Performance**, which refers to the achievable identification criteria (such as accuracy, speed, and robustness), and the required resources to achieve an acceptable identification performance.
- **Acceptability**, that measures to what extend the user may accept the biometric technology in terms of acquisition, data representation, and user privacy. User acceptability will be determined according to the application obtrusiveness and intrusiveness which are related to the user agreement.
- **Circumvention**, is an important parameter that affecting the reliability of the system. It refers to how easy it is to fool the system by fraudulent techniques. The lower circumvention, the better biometric trait [26].

3 Proposed Identification Approach

Analogy to human fingerprints, animal muzzle prints have some discriminative features according to the grooves, or valleys, and beads structures. These uneven features are distributed over the skin surface in the cattle nose area, and they are defined by the white skin grooves or by the black convexes surrounded by the grooves [8]. Return to Fig. 2 for consulting the convexes and the grooves in muzzle prints taken from two different animals.

Minagawa et al. [8] used the joint pixels on the skin grooves as a key feature for muzzle print matching. Some long preprocessing steps were conducted to extract the joint pixels. This approach achieved maximum and minimum matching scores as 60% and 12%, respectively. It achieved unsatisfactory identification performance that was around 30% measured over a database of 43 animals.

Noviyanto and Arymurthy [27] applied Speeded-Up Robust Features (SURF) on muzzle print images for enhancing the identification accuracy. A U-SURF method was applied on 8 animals with 15 images each. The experimental scenario used 10 muzzle pattern images in the training phase, and the other 5 images were used as input samples. The maximum achieved identification accuracy under rotation condition is 90%.

The presented technique in this research is robust from two perspectives. First, it invests the robustness of the SIFT features to image scale, shift, and rotation. Second, it uses the RANSAC algorithm as a robust inliers estimator for enhancing matching results of SIFT features, and ensure the robustness of the matching process. The proposed technique includes SIFT feature extraction, SIFT feature matching, and RANSAC algorithm. Fig. 1 shows a generic and complete muzzle print based identification system, and highlights the cascaded components of the presented approach.

RANSAC algorithm has been developed by Fischler and Bolles [16] especially for computer vision, and it works as a robust estimator. In many images matching cases, RANSAC is an effective robust estimator, which can handle around 50% mismatch contamination levels of the input samples. The integration of the extracted local invariant features and RANSAC is valuable for optimizing the images' similarity score measurement using SIFT features [28].

Admittedly, the generic animal identification system, shown in Fig. 1, works the same way of the human identification one. It has two phases; enrollment

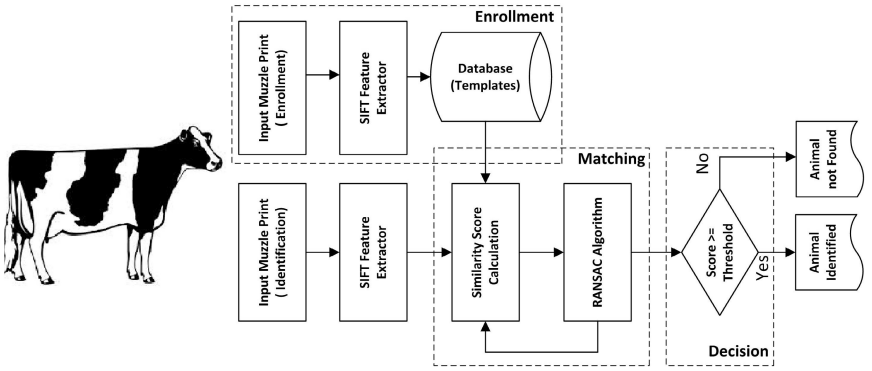


Fig. 1. A flowchart of a complete animal identification system using muzzle print images. The proposed identification approach is represented as a combination between SIFT features and RANSAC algorithm.

phase and identification phase. In the enrollment phase, a muzzle print image is presented, and the SIFT keypoint extractor is applied. Then, the extracted feature vector is stored as a template in the database. The identification phase includes the same enrollment procedure plus matching and decision phases. For calculating the similarity score, the SIFT features of the input image are matched against the templates stored in the database as (1:N) matching approach. The muzzle print image corresponding to the feature vector that has a shortest distance to the input feature vector is considered as the most similar one, and it is given the highest similarity score. RANSAC homography algorithm comes at the end of the matching process to remove any outlier, mismatched SIFT keypoints, data and ensure the robustness of the similarity score. The animal identity is then assigned according to the highest estimated similarity score between the input image and the template one.

4 Experimental Work

The experiments in this paper have been conducted using PC with Intel® core i3-2120 running at 3.30 GHz, and 8 GB of RAM. The PC is empowered by Matlab® and Windows® 64-bit. The VLFeat library [29] has been used for extracting and processing the SIFT keypoints, and it has been installed and optimized for the current experimental environment.

4.1 Muzzle Print Images Database

The lack of a standard muzzle print database was a challenge for conducting this research. Therefore, collecting a muzzle print images database was a crucial decision. The database has been collected from 15 cattle animals with 7 muzzle print images each. A sample of muzzle print images captured from two individual animals is shown in Fig. 2. A special care has been given to the quality

Fig. 2. A sample images of the collected muzzle prints database from live animals. The represented muzzle print images have been taken from two different animals. The muzzle print images show different deteriorates difficulties include orientated images, blurred images, low resolution images, and partial images.

of the collected images. The collected images cover different quality levels and degradation factors such as image rotation and image partiality for simulating real time identification operations.

In identification scenario, 7 images of each animal have been swaped between the enrollment phase an identification phase, and the similarity scores between all of them are calculated. Therefore, similarity score matrix with a dimension of 105×105 have been created. The animal is correctly identified if the similarity scores between the input sample, and the template samples is greater than a specific threshold. Six images of a single animal have been enrolled as templates and marked as $T_1, T_2, T_3, ..., T_6$, and one image has used as input and marked as I_1, S was a similarity function, and H was a similarity score. A correctly identified animal was strictly following the equation as:

$$S(I_1, T_1) \parallel S(I_1, T_2), ..., \parallel S(I_1, T_6) \geq H \tag{1}$$

4.2 Evaluation Results

Preceding to any experimental work, the database images have been processed in terms of image enhancement, image segmentation, and image normalization. The first experimental scenario is directed toward setting the best SIFT parameters that compromise the number of extracted features (keypoints) with the consumed processing time. The preparatory experiments showed that the most effective parameter is the peak threshold (`PeakThresh`) [9], [29], thus the

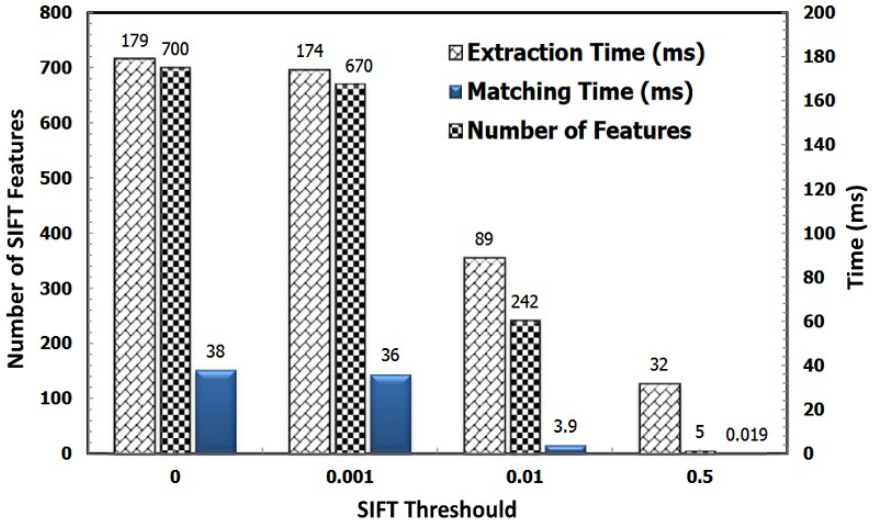


Fig. 3. The behavior the SIFT feature extraction with different peak threshold (`PeakThresh`) values with respect to the number of features, the extraction time, and the matching time.

objective of this scenario is to optimize the peak threshold. The results of the conducted experiments are shown in Fig. 3. The reported results are the average of value of 105 feature extraction processes and 5512 matching operations. The maximum number of features is achieved with (`PeakThresh = 0.0`), however, with (`PeakThresh = 0.001`), the extracted features are reduced by 30, and the extraction time is reduced by 5 ms. The other `PeakThresh` values achieve an unacceptable number of features regardless of the time factor. The optimum `PeakThresh` value is selected as 0.0 seeking for more SIFT features, and hence, more robustness in feature matching. Following on, the SIFT peak threshold is set to that optimum value, whereas the other parameters are kept as defaults.

In the identification scenario, 6 images of each individual animal have been processed and enrolled in the database. The total images in the database were $(6 \times 15 = 90)$, and 1 image has been used as input to simulate the identification operation. According to equation 1, 14 animals out of 15 have been correctly identified which archives equivalent identification accuracy value as 93.3%. It is worth notice that the average consumed feature extraction time is 179 ms and the average individual matching time is 38 ms including RANSAC optimization, which are consistent with Fig. 3. However, both times are considered very short for single feature extraction and matching operation, the total identification time still long, around ≈ 23 s at maximum, because a linear database research method has been used, and the identification time is based on the template location.

5 Conclusions and Future Work

This paper has presented a robust cattle identification approach that uses muzzle print images as input for SIFT feature extraction and matching. Due to the lack of a standard muzzle print database, we have collected 105 images from 15 animals to work as a benchmark for the presented approach. In order to evaluate the robustness of the approach, the collected images cover different deteriorated factors such as rotated images, blurred images, partial images, and low resolution images. The achieved identification accuracy is 93.3% compared to 90% reported throughout the literature. The superiority of the presented technique comes from the coupling of local invariant features with RANSAC homography as a robust outliers removal algorithm. Muzzle print images database extension and standardization for international matchmarking of muzzle print related algorithms are two future directions. The reduction of the identification time in a large database is an interesting challenge that will be tackled as a future work.

Acknowledgement. The authors are very thankful to Dr. Hamdi Mahmoud, Faculty of Computers and Information, BeniSuef University, for the great help of collecting the live captured muzzle print images. The authors are also very thankful to Dr. Eman Hany Hassan and Professor Rabie Hassan Fayed, Faculty of Veterinary Medicine, Cairo University, for the valuable offers in understanding the characteristics of the muzzle print image.

References

1. Vlad, M., Parvulet, R.A., Vlad, M.S.: A survey of livestock identification systems. In: Proceedings of the 13th WSEAS International Conference on Automation and Information, ICAI 2012, pp. 165–170. WSEAS Press, Iasi (2012)
2. Roberts, C.: Radio frequency identification (RFID). Computers & Security 25(1), 18–26 (2006)
3. Jain, A.K., Ross, A.A., Nandakumar, K.: Introduction to Biometrics. Springer (2011)
4. Giot, R., El-Abed, M., Rosenberger, C.: Fast computation of the performance evaluation of biometric systems: Application to multibiometrics. Future Generation Computer Systems 29(3), 788–799 (2013), Special Section: Recent Developments in High Performance Computing and Security
5. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology 14(1), 4–20 (2004)
6. Egawa, S., Awad, A.I., Baba, K.: Evaluation of acceleration algorithm for biometric identification. In: Benlamri, R. (ed.) NDT 2012, Part II. CCIS, vol. 294, pp. 231–242. Springer, Heidelberg (2012)
7. Petersen, W.: The identification of the bovine by means of nose-prints. Journal of Dairy Science 5(3), 249–258 (1922)

8. Minagawa, H., Fujimura, T., Ichiyanagi, M., Tanaka, K.: Identification of beef cattle by analyzing images of their muzzle patterns lifted on paper. In: Proceedings of the Third Asian Conference for Information Technology in Agriculture, AFITA 2002: Asian Agricultural Information Technology & Management, Beijing, China, pp. 596–600 (October 2002)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
10. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2008)
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of 7th IEEE International Conference on Computer Vision, ICCV 1999, Kerkyra, Corfu, Greece, pp. 1150–1157 (September 1999)
12. Iannizzotto, G., Rosa, F.L.: A SIFT-based fingerprint verification system using cellular neural networks. In: Pattern Recognition Techniques, Technology and Applications, pp. 523–536. InTech (2008)
13. Park, U., Pankanti, S., Jain, A.K.: Fingerprint verification using SIFT features. In: Proceedings of SPIE Defense and Security Symposium (2008)
14. Awad, A.I., Baba, K.: Evaluation of a fingerprint identification algorithm with SIFT features. In: Proceedings of the 3rd 2012 IIAI International Conference on Advanced Applied Informatics, pp. 129–132. IEEE, Fukuoka (2012)
15. Chen, J., Moon, Y.S.: Using SIFT features in palmprint authentication. In: Procedings of 19th International Conference on Pattern Recognition, pp. 1–4. IEEE (2008)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM 24(6), 381–395 (1981)
17. Jain, A.K., Bolle, R., Pankanti, S.: Biometrics: Personal Identification in Networked Society, 2nd edn. Springer (2005)
18. Jain, A., Ross, A., Pankanti, S.: Biometrics: a tool for information security. IEEE Transactions on Information Forensics and Security 1(2), 125–143 (2006)
19. Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal 40(3), 614–634 (2001)
20. Lee, Y., Filliben, J.J., Micheals, R.J., Phillips, P.J.: Sensitivity analysis for biometric systems: A methodology based on orthogonal experiment designs. Computer Vision and Image Understanding p. (in press, 2013)
21. Schouten, B., Jacobs, B.: Biometrics and their use in e-passports. Image and Vision Computing 27(3), 305–312 (2009), special Issue on Multimodal Biometrics
22. Awad, A.I.: Machine learning techniques for fingerprint identification: A short review. In: Hassanien, A.E., Salem, A.-B.M., Ramadan, R., Kim, T.-H. (eds.) AMLTA 2012. CCIS, vol. 322, pp. 524–531. Springer, Heidelberg (2012)
23. International Biometric Group: Biometrics market and industry report 2009-2014 (March 2008), `http://www.biometricgroup.com`
24. Li, Y.: Biometric technology overview. Nuclear Science and Techniques 17(2), 97–105 (2006)
25. Luis-Garcia, R.D., Alberola-Lopez, C., Aghzout, O., Ruiz-Alzola, J.: Biometric identification systems. Signal Processing 83(12), 2539–2557 (2003)

26. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer (2009)
27. Noviyanto, A., Arymurthy, A.M.: Automatic cattle identification based on muzzle photo using speed-up robust features approach. In: Proceedings of the 3rd European Conference of Computer Science, ECCS 2012, pp. 110–114. WSEAS Press, Paris (2012)
28. Cheng, L., Li, M., Liu, Y., Cai, W., Chen, Y., Yang, K.: Remote sensing image matching by integrating affine invariant feature extraction and RANSAC. Computers & Electrical Engineering 38(4), 1023–1032 (2012)
29. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/

Algebraic Replay Attacks on Authentication in RFID Protocols

Noureddine Chikouche[1], Foudil Cherif[2], and Mohamed Benmohammed[3]

[1] Department of Computer Science, University of M'sila, Algeria
[2] LESIA Laboratory, University of Biskra, Algeria
[3] LIRE Laboratory, University of Constantine, Algeria

Abstract. One of the most important challenges related to RFID systems is the verification of security proprieties in RFID authentication protocols. Among the important attacks in RFID systems, we speak about the Algebraic Replay Attack on Authentication (ARA). Common characteristic between the verified protocols cannot resist algebraic replay attacks. Our work is articulated on the formal automatic verification of RFID protocols by two different tools, firstly, the Open-source Fixedpoint Model Checker (OFMC) tool, secondary, the Constraint Logic based Attack Searcher (Cl-Atse) tool. These tools sufficient for detecting the attach of type ARA.

1 Introduction

The radiofrequency identification (RFID) systems are steadily becoming paramount due to their vast applications such as supply chain management, mobile phone, health, automated payment systems, e-passport, access control etc. A typical RFID system consists of three entities: (1) the tag (or the label), a small electronic device, supplemented with an antenna that can transmit and receive data, (2) the reader, a device to read and write RFID tags by radio waves and (3) the backend system (or database, server), a centralized place that hosts all data regarding access permissions and which may be consulted by the reader.

One of the most important challenges related to RFID systems is security. The communication channel between the server and the reader is assumed to be secure while the wireless channel between the reader and the tag is assumed to be insecure since it makes it opened to logical attacks on authentication protocol. Among attacks studied in the last years by researchers, we quotes algebraic replay attacks (ARA). The main cause of these attacks is the abuse of the algebraic operator properties employed by the protocols. The operator or-exclusive (xor) is algebraic operator. This operation is used in many RFID protocols and has aroused a lot of interest during the last years; its implementation is low cost and requires some logical gates.

The phases of design and implementation of RFID authentication are important, but the phase of verification of the protocol is very important. To validate the security proprieties (secrecy, authentication, integrity,...etc) of authentication protocol, we use a formal tool of verification. There are several tools of automatic verification of

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 153–163, 2013.

cryptographic protocols. We chose OFMC [1, 2] (Open-source Fixedpoint Model Checker) tool and the CL-Atse (Constraint Logic based Attack Searcher) [3] for the following reasons: they are based on the same specification languages: HLPSL language [4] and AnB language [5]. These tools are the analyzer which models a big number of security protocols (more than 90 protocols). These tools are available using various techniques of validation (Model-checking, Horn Clause, resolution of constraints, rewriting technique).

Our Contribution is articulated around the verification of RFID authentication protocols by using the OFMC and CL-Atse tools after specifying these protocols in specification language. These analyses are based on the automatic verification of three security proprieties: secrecy, tag authentication and server authentication. The verified protocols require one-way function, xor-operator and pseudo-random number generator (PRNG).We prove which of the presented protocols cannot resist algebraic replay attacks.

The rest of this paper is structured as follow: Section 2 presents the specification language, the verification tools and the intruder model. Section 3 presents a RFID authentication protocols. In section 4, we show the verification results and we discuss these results in section 5. Finally, the paper is finished by a conclusion.

2 Formal Automatic Verification

The formal automatic verification of RFID protocols involves the following steps:

- Specification: specification the initial assumptions, the capacity of intruder, the protocol goals (secrecy, authentication, etc.), the roles (the tag and reader), the messages transmitted and the primitives (hash function, PRNG, xor-operator, concatenation, etc.),
- Verification: After verifying the protocol using a validation tool, it is confirmed that the protocol is either safe or it has failed. In case of failure, the tool presents message transmitted between an intruder, a reader and a tag, i.e. describe the trace of attack.

2.1 Intruder Model

Besides modelling security protocols, it is also necessary to model the intruder, that is to say, to define its behaviour and limit. For this, we assume an active Dolev-Yao attacker [6]. This intruder model is based on two important assumptions that are the *perfect encryption* and the *intruder is the network*.

Perfect encryption ensures in particular that: *(1)* an intruder can decrypt a message m encrypted with key k if it has the opposite of that key, *(2)* a key cannot be guessed (during the period of its validity), *(3)* and Given m, it is not possible to find the corresponding ciphertext for any message containing m without knowledge of the key.

The intruder is the network: the intruder has complete control over the network, i.e. it can impersonate a tag, impersonate a reader, obtain any message passing through the network, block or modify messages and it can also derive new ones messages from its initial knowledge and the messages that are received from honest principals during protocol run. The communication between the tag and reader is not assured and based on radio frequencies waves. In this paper, our particular verification gets transmissions on the canal reader - tag only.

For the authentication protocols required or-exclusive operator, other important assumption, an intruder that can exploit the algebraic properties of the XOR operator, which are:

$x \oplus 0 \rightarrow x$	(neutral element)	(1)
$x \oplus x \rightarrow 0$	(nilpotence)	(2)
$x \oplus y \rightarrow y \oplus x$	(commutativity)	(3)
$x \oplus (y \oplus z) \rightarrow (x \oplus y) \oplus z$	(associativity)	(4)

2.2 Protocol Goals

Security proprieties, such as: secrecy, tag authentication, and reader authentication.

- Secrecy: or confidentiality, the verification of secret data so that they are never passed on clearly to air on the radio frequency interface which can be spied on.
- Tag authentication: A reader has to be capable of verifying a correct tag to authenticate and to identify this tag in complete safety.
- Reader authentication: A tag has to be capable of confirming that it communicates with the legitimate reader (we assume the communication between the server and reader is assured).

2.3 Specification

In this paper, we use two specification languages, HLPSL [4] and AnB [5]. These languages are the input languages of the OFMC and the CL-Atse verification tools. Alice and Bob (AnB) notation is a high-level and straight-forward language for describing security protocols. It describes how messages are exchanged between honest agents acting in the different protocol roles. The novel features of AnB are its support for protocols that require algebraic properties for the protocol execution, as well as a notion of several types of communication channels that can be used both as assumptions and as goals of a protocol.

High Level Protocol Specification Language (HLPSL) is a modular, expressive, formal, role-based language. Protocol specification consists of two types of roles, basic roles and composed roles. Basic roles serve to describe the actions of one single agent in the run of the protocol. Others instantiate basic roles to model an entire protocol run, a session of the protocol between multiple agents, or the protocol model itself.

2.4 Verification Tools

The OFMC and CL-Atse are developed in the framework of the AVISPA European Project[1] and the AVANTSSAR European Project[2]. These tools can verify the protocols requiring the operator exclusive or (XOR). The first tool, The Open-source Fixedpoint Model Checker (which extends the on the-Fly Model Checker, the previous OFMC) [1, 2] consists of two modules. The classical module performs verification for a bounded number of transitions of honest agents using a constraint-based representation of the intruder behavior. The fixed point module allows verification without restricting the number of steps by working on an over-approximation of the search space that is specified by a set of Horn clauses using abstract interpretation techniques and counterexample-based refinement of abstractions. Running both modules in parallel, OFMC stops as soon as the classic module has found an attack or the fixed point module has verified the specification, so as soon as there is a definitive result.

The second tool, CL-AtSe [3] is a Constraint Logic based Attack Searcher for the security protocols and services takes as an input a service specified as a set of rewriting rules, and applies rewriting and constraint solving techniques to model all states that are reachable by the participants and decides if an attack exists with respect to the Dolev-Yao intruder.

3 RFID Authentication Protocols

In this section, we describe an authentication protocols in RFID system, the common characteristic between these protocols: (i) they use or-exclusive operator and one-way function in transmitted messages and (ii) the vulnerabilities of these protocols are of type algebraic replay attacks on authentication (ARA).

To describe informally many RFID authentication protocols, we afterward, use the following notations:

T	RFID tag or transponder
R	RFID reader or transceiver
H	One-way hash function
‖	Concatenation of two inputs
ID	The unique identifier of a tag
⊕	Or-exclusif
RID	The unique identifier of a reader
S,x,y	Secret value
RH	Right-half of the message
LH	Left-half of the message
Nr,Nt, N_{db}	Random number
CRC	Cyclic Redundancy Check

[1] http://www.avispa-project.org
[2] http://www.avantssar.eu

Table 1. RFID Authentication Protocols

PROTOCOL	**Auth_Tag**	**Auth_Reader**	**Secret Data**	**α**	***f***
LAK [7]	H(Nr⊕Nt⊕K)	H(H(Nr⊕Nt⊕K)⊕K⊕Nr)	K	K	H
CH [8]	LH(RH(ID)⊕h(Nr ⊕Nt⊕K))	RH(RH(ID)⊕h(Nr⊕Nt⊕K))	ID, K	K	H
YL [9]	x⊕h(h(K)⊕nt), h(y⊕Nr⊕ Nt)	y^*⊕h(x^*⊕y), h(x^*⊕y^*)	h(k),y,x,K	y	H
QYY [10]	CRC(ID⊕Nt⊕Nr), CRC(ID⊕ Nt⊕Nr) ⊕ x	CRC(ID⊕Nt), CRC(ID⊕Nt) ⊕ x	ID	ID	CRC
WHC [11]	H(Nr⊕Nt⊕S)	H(ID⊕N_{db})	S, ID	S	H

We can describe the transmitted messages in RFID mutual authentication protocols in form:

```
R → T : Nr
T → R : Nt, Auth_Tag
R → T : Auth_Reader
```

The transmitted messages of `Auth_Tag` and `Auth_Reader` are presented in table 1. The `Auth_Tag` comprises of $f(\alpha \oplus N_t \oplus N_r)$, with α is secret data shared between the tag and reader and f is one-way function such as hash function and CRC function. The following is a detailed description of each step of these protocols:

- The reader RFID produces a nonce Nr and sends it and a request to the tag.
- After receiving Nr, a tag generates a random number Nt and computes the function Auth_Tag, then sends Nt. The Auth_Tag is back to the reader (server).
- After receiving authentication message from the tag, the reader would search whether there exists certain ά in table α of the database, which could make f(ά ⊕Nt⊕Nr)= f(α⊕Nt⊕Nr). If it is found, the tag crosses the authentication of the tag and is considered as legitimate, and then the reader calculates Auth_Reader, then sends Auth_Reader to the tag.
- The tag computes Auth_Reader‘, If the outcome equals to the received Auth_Reader, the authentication of the reader is successful, otherwise, the authentication has failed.

Our paper verifies five protocols, as following:

- LAK [7]: Lee et al. propose an authentication protocol. The reader R and tag T share secrets k. at finish authentication, reader and tag updates k to h(k).
- CH [8]: The CH protocol is proposed by Chien and Huang in 2008. It uses hash function and primitives non-cryptographic (Left, Right and Rotate). It uses these primitives for increase the security of protocols.

- YL [9]: The author Yanfei Liu provides a detailed security analysis of the protocol and claims that YL achieves a list of security properties, including resistance to tag impersonating, denial of service, replay and compromising attacks.
- QYY [10]: The authors of this protocol claim that this protocol is secure because of the use CRC (Cyclic Redundancy Check) and uses random nonces to encrypt messages.
- WHC: Wei et al. [11] proposed an authentication protocol (WHC protocol) in 2011. The server and tag share secrets value S and Identifier ID, this protocol proposed for application RFID-Mobile.

4 Results of Verification

This section is articulated around the verification of LAK protocol (as an example) by using CL-Atse and OFMC tools after having specified this protocol in HLPSL and AnB languages respectively.

4.1 OFMC Result

OFMC tool detects the trace of attack on RFID tag authentication (see Fig. 1 (a)). In this trace result, `i` represents the intruder, `(x501, 1)` the reader (server), and `(x502,1)` the tag. The posted information such as: `NR(1)` is the instance of the nonce `NR`, `X2624` and which is a variables related to the internal workings of the OFMC tool (in this trace is the instance of the nonce `NR`), `NT(2)` is the instance of the nonce `Nt` and `sk(x502,x501)` is a symmetric key `K`.
We symbolize: `NR(1)` by `Nr, X2624` by `Nr'`, `NT(2)` by `Nt and sk(x502,x501)` by `K`. We Summarizes this trace as the following:

```
(1) R → I : Nr
(2) I → T : Nr⊕Nr'
(3) T → I : Nt, H(Nr'⊕Nt⊕K)
(4) I → R : Nr'⊕Nt, H(Nr'⊕Nt⊕K)
(5) R → I : H (H(Nr⊕Nr'⊕Nt⊕K)⊕K⊕Nr)
```

Several comments can be drawn from the trace:

- *Msg1:* The reader generates a nonce Nr and the intruder captures and stores the nonce in the course of the communication.
- *Msg2*: The intruder generates another nonce Nr' and sends Nr⊕Nr' to the tag.
- Msg3: The tag generates an instance of the nonce Nt and sends it with the hash function h(K⊕Nr⊕Nr'⊕Nt) to the intruder.
- Msg4: The intruder returns the received function to the reader with Nr'⊕Nt.
- Msg5: The reader sends the message h(h(K⊕Nr⊕Nr'⊕Nt)⊕K⊕Nr)to the tag. This message does not depend on the discovered attack (Impersonation of tag).

The attack on RFID tag authentication is realised in Msg4. We will describe the principle of this attack in the section of discussion.

```
% Open-Source Fixedpoint Model-Checker
   version 2012c
INPUT
   Lak.AnB

SUMMARY
  ATTACK_FOUND
GOAL:  weak_auth
DETAILS
BACKEND  OFMC
STATISTICS
  TIME 2184 ms
  parseTime 0 ms
  visitedNodes: 9 nodes
  depth: 2 plies

ATTACK TRACE
(x501,1) -> i: NR(1)
i -> (x502,1): NR(1) XOR x2624
(x502,1) -> i: NT(2),hash(sk(x502,x501)
   XOR NR(1) XOR x2624 XOR NT(2))
i -> (x501,1): x2624 XOR
   NT(2),hash(sk(x502,x501) XOR NR(1)
   XOR x2624 XOR NT(2))

(x501,1) -> i: hash(hash(sk(x502,x501)
   XOR NR(1) XOR x2624 XOR NT(2)) XOR
   sk(x502,x501) XOR NR(1))
% Reached State:
request(x501,x502,pRTNThashXorXorskTRNRN
   T,x2624 XOR NT(2),hash(sk(x502,x501)
   X
OR NR(1) XOR x2624 XOR NT(2)),1)
% state_rR(x501,2,sk(x502,x501),
   hash,x502,NR(1),hash(sk(x502,x501)
   XOR NR(1) XOR
 x2624 XOR NT(2)),x2624 XOR NT(2),x2624
   XOR NT(2),hash(sk(x502,x501) XOR
   NR(1) X
OR x2624 XOR
   NT(2)),hash(hash(sk(x502,x501) XOR
   NR(1) XOR x2624 XOR NT(2)) XOR x
2624 XOR NT(2)),1)
% state_rT(x502,1,hash,
   sk(x502,x501),x501,NR(1) XOR
   x2624,NT(2),NT(2),hash(sk(x5
02,x501) XOR NR(1) XOR x2624 XOR
   NT(2)),1)
%
   witness(x502,x501,pRTNThashXorXorskTR
   NRNT,NT(2),hash(sk(x502,x501) XOR
   NR(1) X
OR x2624 XOR NT(2)))
```

(a) OFMC Result

```
%% Constraint Logic-based ATtack SEarcher
(CL-ATSE) Version 2.5-18 (2012-septembre-
26).
   ---------------
     AtSe Summary
   ---------------
     Protocol file:
     Attack found : YES
     Analysed   : 6 states
     Reachable  : 3 states
     Translation: 0.00 seconds
     Computation: 0.00 seconds
   ---------------------------------------
                  Attack Description
(the list of protocol steps followed by
cl-atse)
   ---------------------------------------
   Short attack description :
   -------------------------
    Kind of attack:  Authentication on
(r,t,aut_tag,xor(X4064,n1(Nr),n5(Nt)))
       UnivQ. Vars:   false
      Substitution:  [
Nt(2)=xor(X4064,n1(Nr),n5(Nt))  Nr(5)=X4064
]
     Compact trace:   (r,3) (t,4) (r,3)
   Detailed attack description :
   ----------------------------
    i -> (r,3):  start
    (r,3) -> i:  n1(Nr)
                 &
Witness(r,t,aut_reader,n1(Nr));
                 & Built from step_0
    i -> (t,4):  X4064
    (t,4) -> i:
n5(Nt).{xor(X4064,k,n5(Nt))}_h
                 & Secret(k,(),set_68);
Witness(t,r,aut_tag,n5(Nt));
                 & Built from step_2
    i -> (r,3):
xor(X4064,n1(Nr),n5(Nt)).{xor(X4064,k,n5(N
t))}_h
    (r,3) -> i:
{xor(k,n1(Nr),{xor(X4064,k,n5(Nt))}_h)}_h
                 & Secret(k,(),set_60);
                 &
Request(r,t,aut_tag,xor(X4064,n1(Nr),n5(Nt
)));
                 & Built from step_1
   %% Job terminated successfully.
```

(b) CL-Atse Result

Fig. 1. Traces attacks on the LAK protocol

4.2 CL-Atse Result

Fig 1. (b) and Fig 2. shows the trace of attack on LAK protocol with the CL-Aste tool. We Summarizes this trace as the following:

(1) I → R : start
(2) R → T : Nr

(3) I → T : Nr‘
(4) T → I : Nt, H(Nr‘⊕Nt⊕K)
(5) I → R : Nr‘⊕Nr⊕Nt, H(Nr‘⊕Nt⊕K)
(6) R → I : H (H(Nr‘⊕Nt⊕K)⊕K⊕Nr)

The principle of the detected attack on LAK protocol by OFMC and CL-Atse is the same. The only difference is that intruder in CL-Atse generates a Nr’ nonce and sends it to the tag, but the intruder in OFMC generates the same nonce Nr and computes the xoring of Nr‘ with Nr (e.g. Nr⊕Nr‘), and sends the result to tag.

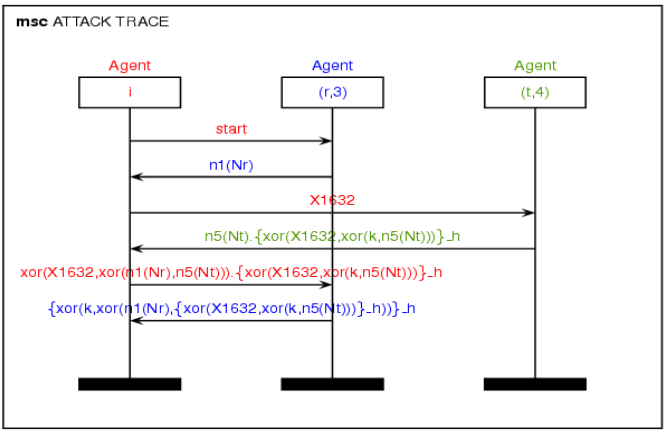


Fig. 2. Message Sequence Chart of ARA (Cl-Atse)

5 Discussion

In this section, we analyze the results of RFID authentication protocols and we quote the implementation and the countermeasure of ARA attacks.

Our results are based on the automatic verification of the authentication properties of each RFID authentication protocol. Concerning the message of tag authentication Auth_tag, the difference between these protocols is the type of one-way function (hash function and CRC) and the secret data which shared between the tag and the reader (server).

For tag impersonation of the studies protocols, an intruder can store all the messages exchanged in a protocol run. To tag impersonate, the adversary could replay f(α⊕nr⊕nt) if he ensures that f(α⊕nr⊕nt) = f(α⊕nr’⊕nt’). The activate intruder can generate a new none and make an algebraic calculate of the type xor operation between numbers. Then, to satisfy this condition the intruder sets nt’ to nr⊕nr’⊕nt. Here is the detail:

f(α⊕nr⊕nt) =? f(α ⊕nr’⊕nt’)

f(α⊕nr⊕nt) =? f(α ⊕nr’⊕nr⊕nr’⊕nt) ➔ replace nt’

$f(\alpha \oplus nr \oplus nt) = ?\ f(\alpha \oplus nr' \oplus \underline{nr' \oplus nr \oplus nt})$ ➔ commutativity

$f(\alpha \oplus nr \oplus nt) = ?\ f(\alpha \oplus 0 \underline{\oplus nr \oplus nt})$ ➔ nilpotence

$f(\alpha \oplus nr \oplus nt) = f(\alpha \oplus nr \oplus nt)$ ➔ neutral element

All the studied protocols cannot resist RFID tag authentication attack, and therefore an intruder can impersonate the tag. This type of attack is based on algebraic properties of algebraic operators (or, and, xor). The paper [12] aims to identify the algebraic problems which enable many attacks on RFID protocols. Toward this goal, three emerging types of attacks on RFID protocols, concerning authentication, untraceability, and secrecy are discussed. The common theme in these attacks is the fact that the algebraic properties of operators (e.g. *xor* operator) employed by the protocols are abused. The methods used to find algebraic replay attacks are sufficiently straight-forward. The algebraic replay attacks in RFID authentication protocols are described in some works such as [13, 14, 15, 16, 17, 18].

The relay attack system can use two transponders in order to relay the information that a reader and a token exchange during a cryptographic challenge response protocol. A proxy-token device is placed near the real reader and a proxy-reader device is placed near the real token, possibly unknown to its holder. Information can therefore be forwarded over a great distance if a suitable communication medium is chosen between the proxy-token and proxy-reader. As a result, the reader will report that it has verified the presence of a remote token and provide access to the attacker [19]. Practically, the ARA system is based on relay attack system. The difference between this system and relay attack system is: this system supports Dolev-Yao attack model (see section 2). Therefore, the proxy system can generate a random number and compute xor operation between numbers. The process of attack system for LAK protocol as following (see figure 3):

1. Legitimate reader generates a nonce Nr and sends it to the proxy-token.
2. Proxy-token receives it and blocks it; the proxy-token generates a nonce Nr' and forwards this nonce to the proxy-reader through the fast communication channels.
3. Proxy-reader fakes the real reader, and sends Nr' to the legitimate tag.
4. Legitimate tag computes a new nonce Nt and computes hash function H(Nr'⊕Nt⊕K) and transmits it to the proxy-reader.
5. Proxy-reader receives it and calculates the new nt' = nr⊕nr'⊕nt and forwards this message and hash function received to the proxy-token through the fast communication channel.
6. Proxy-token forwards nt' and H(Nr'⊕Nt⊕K) to the real reader.

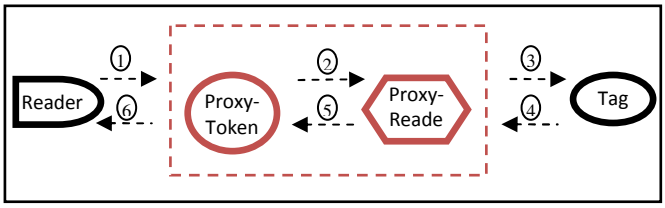


Fig. 3. Attack System

The principal vulnerability in studies protocols in use of xor operator in one-way function. Consequently, the solution is to change the primitive XOR (⊕) between the values of one-way function (α, Nr,Nt) by the concatenation (‖). Therefore, the new one-way function is: $f((\alpha \oplus Nr) \| Nt)$ or $f(\alpha \| (Nr \oplus Nt))$.

6 Conclusion

We have presented in this paper different protocols using xor-operator and one-way functions. The one-way functions in studying protocols are: hash function and CRC function. Our security analysis of these RFID authentication protocols by automatic formal tools. We showed that the verified protocols cannot resist RFID tag authentication attack therefore; an intruder can impersonate the tag.

The detected attack is the type of algebraic replay attacks (ARA) on tag authentication. The principal cause of the described attacks in our work is the abuse of the proprieties of xor-operator in the transmitted messages. The proposed solution for this attack is correcting the use of xor-operator and replacing it by concatenation operator.

References

1. Basin, D., Mödersheim, S., Viganò, L.: OFMC: A symbolic model checker for security protocols. International Journal of Information Security 4(3), 181–208 (2005)
2. Modersheim, S., Vigano, L.: The open-source Fixed-point model checker for symbolic analysis of security protocols. In: Aldini, A., Barthe, G., Gorrieri, R. (eds.) FOSAD 2007/2008/2009. LNCS, vol. 5705, pp. 166–194. Springer, Heidelberg (2009)
3. Turuani, M.: The CL-Atse Protocol Analyser. In: Pfenning, F. (ed.) RTA 2006. LNCS, vol. 4098, pp. 277–286. Springer, Heidelberg (2006)
4. HLPSL Tutorial: A Beginner's Guide to Modelling and Analysing Internet Security Protocols (2005), http://www.avispa-project.org/
5. Mödersheim, S.: Algebraic Properties in Alice and Bob Notation. In: Proceedings of Ares 2009, pp. 433–440. IEEE Xplore (2009); Extended version: Technical Report RZ3709, IBM Zurich Research Lab (2008)
6. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory IT-29(2), 198–208 (1983)
7. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Symposium on Cryptography and Information Security (2006)
8. Chien, H.-Y., Huang, C.-W.: A lightweight RFID Protocol Using Substring. In: Embedded Ubiquitous Computing (EUC), pp. 422–431 (2007)
9. Liu, Y.: An Efficient RFID Authentication Protocol for Low-Cost Tags. In: Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, pp. 706–711 (2008)
10. Qingling, C., Yiju, Z., Yonghua, W.: A minimalist mutual authentication protocol for RFID system & BAN logic analysis. In: Proc. of CCCM 2008, pp. 449–453. IEEE Computer Society, Los Alamitos (2008)
11. Wei, C.-H., Hwang, M.-S., Chin, A.-Y.: A Mutual Authentication Protocol for RFID. IT Professional 3, 20–24 (2011)

12. van Deursen, T., Radomirović, S.: Algebraic Attacks on RFID Protocols. In: Markowitch, O., Bilas, A., Hoepman, J.-H., Mitchell, C.J., Quisquater, J.-J. (eds.) WISTP 2009. LNCS, vol. 5746, pp. 38–51. Springer, Heidelberg (2009)
13. Cao, T., Shen, P.: Cryptanalysis of Two RFID Authentication Protocols. International Journal of Network Security 9(1), 95–100 (2009)
14. Jannati, H., Falahati, A.: Cryptanalysis and Enhanced of Two Low Cost RFID Authentication protocols. International Journal of UbiComp 3(1), 1–9 (2012)
15. van Deursen, T., Radomirovic, S.: Attacks on RFID Protocols. Report 2008/310, Cryptology ePrint Archive (2008)
16. Chen, X., van Deursen, T., Pang, J.: Improving Automatic Verification of Security Protocols with XOR. In: Breitman, K., Cavalcanti, A. (eds.) ICFEM 2009. LNCS, vol. 5885, pp. 107–126. Springer, Heidelberg (2009)
17. Mihailescu, M.I.: Resreach on Solutions for Preventing Algebraic Attacks Against Biometric and RFID Protocols. ACTA Universitatis Apulensis (Special Issue), 371–386 (2011)
18. Chikouche, N., Cherif, F., Benmohammed, M.: Vulnerabilities of two Recently RFID Authentication Protocols. In: International Conference on Complex Systems, Agadir, Morocco (2012)
19. Hancke, G.P.: Practical Attacks on Proximity Identification Systems. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 328–333 (May 2006) (short paper)

A Privacy Preserving Approach to Smart Metering

Merwais Shinwari[1], Amr Youssef[1], and Walaa Hamouda[2]

[1] Concordia Institute for Information Systems Engineering (CIISE)
[2] Electrical and Computer Engineering Department
Concordia University
Montreal, Canada
(m_shinwa,youssef)@ciise.concordia.ca, hamouda@ece.concordia.ca

Abstract. High frequency power consumption readings produced by smart meters introduce a major privacy threat to residential consumers as they reveal details that could be used to infer information about the activities of home occupants. In this paper, we question the need to disclose high frequency readings produced at the home's level. Instead, we propose equipping smart meters with sufficient processing power enabling them to provide the utility company with a set of well-defined services based on these readings. For demand side management, we propose the collection of high frequency readings at a higher level in the distribution network, such as local step-down transformers, as this readily provides the accumulated demand of all homes within a branch. Furthermore, we study the effect of the proposed approach on consumers' privacy, using correlation and relative entropy as measures. We also study the effect of load balancing on consumers' privacy when using the proposed approach. Finally, we assess the detection of different appliances using high frequency readings collected for demand side management purposes.

Keywords: Smart Grid, Smart Meter, Privacy, Advanced Metering Infrastructure (AMI).

1 Introduction

The electric grid in use today is undergoing a transformation to improve its efficiency and reliability through the use of computation and communication technologies. A major part of this is the enhancement of the distribution network through the introduction of smart meters. These meters collect power consumption readings and transmit them to the utility company in an automated way. Unlike traditional meters, smart meters produce detailed chronological high frequency readings that reveal both the time of consumption and the amount of power consumed. Utility companies argue that high frequency readings form the basis for time of use billing schemes, which are expected to cause consumers to shift part of their consumption to off-peak hours resulting in a flat demand profile (i.e., one with a small peak-to-average ratio) and improving energy production and consumption efficiencies [1]. Furthermore, it is argued that a clearer vision of the distribution network helps in improving service quality and reliability. For example, a utility company could centrally detect and respond to

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 164–176, 2013.

blackouts and brownouts more effectively in comparison to user initiated notifications [2]. Because of this, smart metering is viewed as the fundamental platform that facilitates service enhancement, reliability, and efficiency and is considered an enabler for technologies such as proactive energy consumption management and load balancing techniques [3] [5].

By their nature, electrical appliances consume power in specific patterns which produce detectable signatures. For example, a standard incandescent lamp constantly consumes a fixed amount of power during its operation period, whereas a refrigerator consumes most of its power during its cooling cycles, when the compressor is running, and significantly less power during its idle cycles. Such patterns can be used to produce signature libraries which can be used for appliance detection and identification [6] [7].

Given a library of power consumption signatures of appliances, and the detailed power consumption of a home, this home's consumption can be decomposed and individual appliances can be detected using Nonintrusive Appliance Load Monitoring technologies (NALM) [8]. Fig. 1 shows an example of appliance detection using power signatures. As shown, many appliances can be identified through their distinct power consumption patterns. That is, with NALM technologies, high frequency readings produced by smart meters offer a window into the activities of homes' occupants. This includes the identification of appliances and any other information possibly inferable from the appliances used. Furthermore, by observing the real time power consumption of a given home, an intruder can identify when the occupants are awake/asleep or whether the home is occupied or not [9].

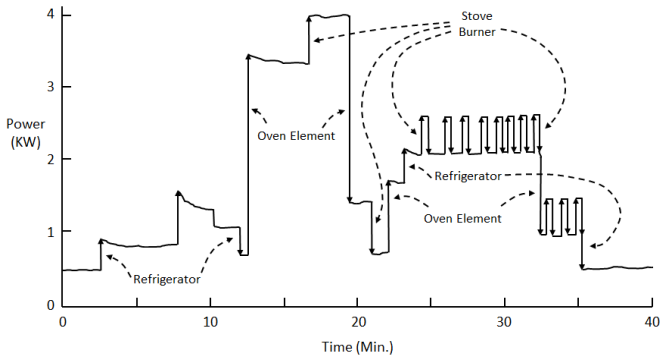


Fig. 1. Appliance identification through power consumption patterns [8]

Although high frequency readings produced by smart meters enable improving the efficiency of the electric grid, they introduce a privacy threat that was not present in the classical grid. In this paper, we present an alternative approach to smart metering with the objective of maintaining its advantageous functionality while preserving consumers' privacy. In the following section, we briefly review some related work in this area. Our approach to smart metering is explained in section 3. In section 4, we assess the privacy gained from the proposed approach. Finally, in section 5, we present our conclusion.

2 Related Work

The privacy impact of collecting power consumption readings in high frequency is well known and widely studied. Reports such as [10], [11], and [12] indicate that the privacy concerns of smart metering must be taken into consideration and addressed at the design stage rather than as a later addition. Furthermore, many researchers have proposed various approaches to address this problem. In this section, we present a selection of the main contributions in this area highlighting the core ideas proposed.

Kalogridis et al. [13] propose masking the power signature of appliances using a rechargeable battery. In particular, the authors propose the use of an energy routing device that controls power flow from the grid to the home and from/to a rechargeable battery following a water-filling based algorithm. This either charges or discharges the battery in a way that masks some details of the home's power demand. Additional work presented in [14] attempts to quantify the privacy offered by the battery solution, and concludes that privacy preservation increases as the battery size gets larger. Although this approach does mask part of the consumption profile, introducing a large rechargeable battery and a power routing device presents a hindrance to consumers. Furthermore, this solution does not offer as much privacy as consumers had before the introduction of smart meters. For example, information deducible from a home's general consumption pattern such as "when did the occupants wake up" or "is the house occupied" can still be attained even with the deployment of this solution.

Efthymiou and Kalogridis [15] argue that although high frequency readings may be needed for operational purposes, there is no need to attribute them to specific consumers. Consequently, the authors propose the use of two sets of readings: one in high frequency, and the other in low frequency. The high frequency readings are to be collected anonymously with the help of an escrow service and are provided to the utility company. Since the consumers' identities are not associated with these records, the consumption and usage characteristics cannot be traced to a specific consumer. The lower frequency readings are to be bounded to their respective consumers and used for billing purposes. Since these do not capture detailed power consumption information, they are not a threat to consumers' privacy. Although this method may seem effective, the use of an escrow service simply transfers the trust problem from the utility company to the escrow service provider, and therefore, does not provide a fundamental solution to the original problem.

In [16], Tomosada and Sinohara propose that smart meters transmit synthetically produced data that shares the same statistical properties of the real readings instead of transmitting the readings themselves. The authors argue that since this virtual demand shares the same statistical properties with the real demand, it can be used for demand side management when averaged over multiple users. In their work, the authors propose a methodology for producing virtual demand from the real demand and conclude that this approach preserves the consumer's privacy. Although this method produces correct statistics, other characteristics critical to demand side management could be lost, for example, the peak value and the time at which this peak value occurs.

3 Proposed Metering Approach

Smart meters are typically used as distributed data acquisition devices. That is, the meters only produce and transmit high frequency readings to the utility company. The utility company, in turn, centrally processes this data producing bills for its subscribers based on the time power was consumed. Fig. 2 illustrates this view of smart meters functionality.

With this approach, high frequency readings are present at the meter, in transmission and in storage at the utility company's processing facility. Having this data at all these points maximizes the potential attack surface for an attacker. This way, the attacker needs to identify some vulnerability in any of these points to be able to access the detailed consumption records. Furthermore, if an attacker is able to identify and exploit some vulnerability at the central processing facility, the impact would be devastating as hundreds of thousands of records could be compromised in a single breach. Verifiably securing large distribution networks, communication networks and processing facilities is practically infeasible. Furthermore, an attempt to secure such interconnected systems would be a tedious task that is almost impossible to implement flawlessly.

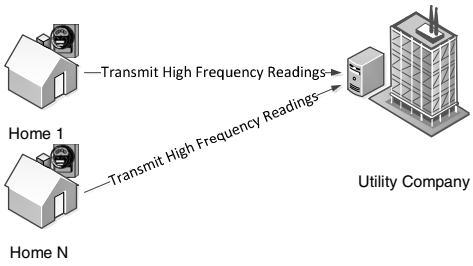


Fig. 2. Smart meters as data acquisition devices. Meters produce and transmit high frequency readings to the utility company

For the purpose of our work, we categorize consumer-oriented data collected by the utility company into two basic types, namely subject data and community data. We define subject data as data collected from and identifiable to a single consumer. We assume that actions taken based on this type of data will only affect its associated consumer. On the other hand, we define community data as data identifiable to a group rather than a single consumer. Furthermore, we assume that actions taken on a large scale, i.e., on the community as a whole, are based on this type of data.

We propose the deployment of smart meters in a way that segregates these two types of data offering each an appropriate level of protection. Therefore, from a privacy perspective, subject data would have a higher level of protection in comparison to community data. To do so, we propose the use of two sets of meters positioned at different locations in the distribution network, namely home meters and zone meters.

3.1 Home Meters

We propose that home meters function as service providing modules rather than data acquisition devices. That is, assuming that each meter is equipped with sufficient

processing power, the meters are to offer the utility company a set of well-defined services computed over the consumer's high frequency readings. In addition, home meters are not to disclose the collected high frequency readings to any party. Furthermore, the services offered by the meter must be developed on a need to know basis.

With this approach, the meters become the entities that perform all required processing on their respective consumers' data and only the outcomes of the processing, i.e., the final results, are made available to the utility company. This allows home meters to provide the desired functionality while eliminating the need to disclose users' high frequency readings, consequently, preserving the consumers' privacy. Furthermore, this introduces a point of control on the type of information the utility company gains access to.

The services provided by a home meter would depend on the protocols/functions it implements. For example, for billing purposes, meters would implement a billing protocol that starts by receiving an authenticated request from the utility company to produce the consumer's bill for a given period. The meter, in turn, uses the high frequency readings from its internal storage to compute the amount owed in dollars based on a pre-agreed upon pricing scheme. The final result of the process would be encrypted and digitally signed by the meter and transmitted to the utility company. This would provide the utility company with the desired information ensuring that it was produced by the meter.

Besides billing, other useful functionalities can be easily implemented. For example, meters could report their operation status or fault codes by periodically transmitting a status message that can be protected using cryptographic techniques. Fig. 3 illustrates the use of home meters as service providing modules.

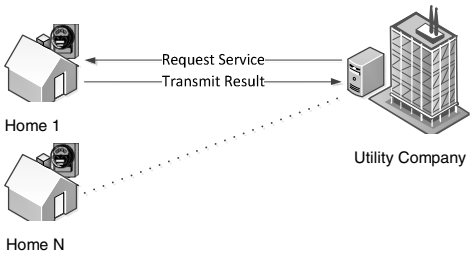


Fig. 3. Home meters as service providing modules. Utilizing the high frequency readings, home meters offer the utility company a set of well-defined services on a need to know basis.

It should be noted that, in our proposed approach, home meters must be trusted as they are the only components that hold and directly process their respective consumer's high frequency power consumption readings. Since a meter is a relatively small device that implements a limited set of protocols, it is feasible to build verifiably secure meters. This can be achieved through the use of trusted computing platforms [17] [18], and the use of formally verifiable designs.

With this approach, since each consumer's data is only present at a single point, the attack surface is significantly reduced. That is, if an attacker is successful in penetrating

a given meter, only the records belonging to the associated customer would be compromised. This is a significant security advantage in comparison to the use of meters as distributed collection nodes and centrally processing the data collected.

3.2 Zone Meters

As stated above, home meters do not disclose unprocessed measurements of power consumption; rather they provide specific information computed from this data. Even though it is possible to produce accurate time of use based consumption bills, information on consumption trends and the time of power consumption would be masked. This introduces a hindrance to the operations of utility companies as such information is aggregated for demand side management. To address this, we propose the use of an additional small set of meters placed at a higher level in the distribution network; typically at local step-down transformers.

Instead of securely producing accumulative readings through an escrow service or using cryptographic approaches, we take advantage of the already existing topology of the distribution network. By observing that the demand at a step down station is the accumulation of the demand of all homes supported by this station, measuring the power consumption at this level is equivalent to aggregating the power demand of individual homes in this zone. Therefore, by collecting measurements at this location using zone meters, we readily obtain the accumulative demand of all homes within a given branch. Fig. 4 illustrates the use of two sets of meters at the home and the zone level.

Although high frequency readings produced by zone meters are not directly attributable to a single consumer, they produce readings of the composite demand of all homes supported by their branch which is a function of the consumption of each home. In the following section we analyze the privacy impact of the proposed approach and the visibility of a home's demand through readings produced by zone meters. We also specifically consider the impact of load balancing on consumers' privacy when using the proposed approach.

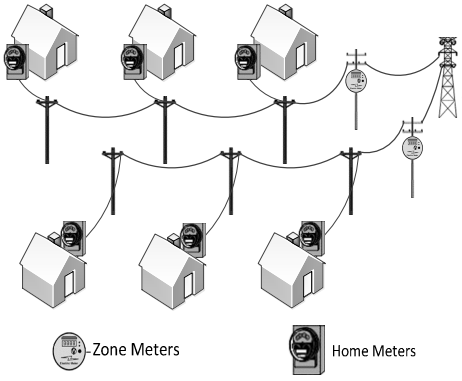


Fig. 4. The use of meters at different levels in the distribution network. Home meters offer a set of well-defined services at the consumer level whereas zone meters produce high frequency readings at the neibghorhood level.

4 Privacy Assessment

In our proposed approach, high frequency readings produced by home meters are not disclosed to the utility company or any other party. This eliminates the direct privacy impact introduced by these readings. Although zone meters introduce indirect threats, the superposition of signatures from different homes introduces an obfuscation effect on individual signatures which lessens the disclosed information. In this section, we assess the obfuscation gained by overlapping home signatures. The number of homes supported by a single zone meter is a main factor in the level of signature overlap; therefore we consider this as a primary factor in our simulation. Furthermore, since time of use billing schemes are expected to flatten the overall demand of a community which affects the overlapping of signatures, we take this into consideration as well.

4.1 Simulation Environment

We produced our simulation environment using a set of appliances with distinct power signatures similar to [8] and using measurements from [19]. Each home is allocated a set of appliances and the operation time of each appliance is selected randomly. The simulation is conducted over a period of 24 hours. Furthermore, to assess the impact of load balancing, two sets of results are produced for each simulation scenario. The first represents the case where power is consumed at will. This type of consumption results in the appearance of peak demand hours as is the case with the classical power grid [20]. The second represents the case where consumers shift part of their consumption to off-peak hours using load balancing techniques such as those described in [3], [4], and [5]. This results in a relatively flat consumption profile for the community as a whole. Fig. 5 shows the simulated power demand for a sample home. As depicted in the figure, the consumption patterns of many appliances can be easily identified.

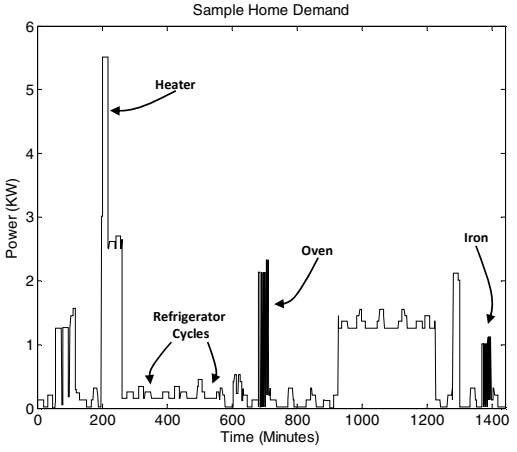


Fig. 5. Simulation of the demand of a single home with a sampling interval of one minute

To simulate the readings produced by zone meters, a variable number of homes are simulated and accumulated. Fig. 6 illustrates the aggregate demand of a community of 50 homes. As depicted in the figure, the overlapping of signatures of different homes distorts the appliance signatures. The figure also reflects the effect of load balancing on the overall consumption of the community. As shown, the use of load balancing results in a flatter overall demand with a lower peak to average ratio. This results in a more uniform level of overlapping between appliance signatures.

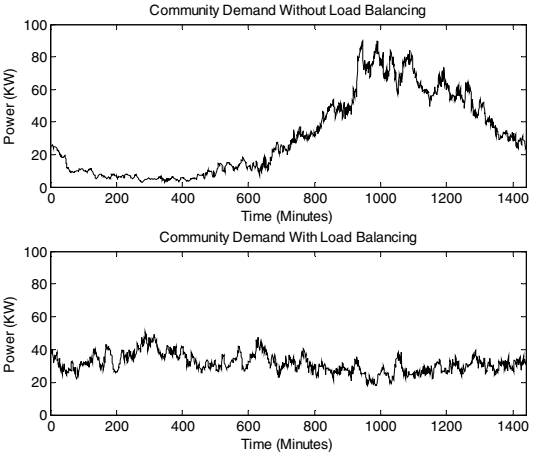


Fig. 6. Community demand without load balancing (top), and with load balancing (bottom)

Fig. 7 shows the decrease in correlation between a home's demand and the overall community consumption as the number of homes in the community is increased. The decreasing trend in the figure can be attributed to the distortion caused to the demand signature from other homes. As the number of homes increase, so does the level of distortion. Furthermore, the figure shows that the use of load balancing techniques further reduces the correlation value resulting in better privacy. This is because load balancing results in even overlapping and eliminates the low overlap between signatures during low demand hours.

Fig. 8 shows the correlation value in the average case for each home when simulating a community of 50 homes. The results indicate that the signatures of all simulated homes were distorted to a similar level.

As an assessment of the difference between the probabilistic distributions in the data sets, we compute the Kullback Leibler divergence (also known as the relative entropy) while increasing the number of homes in the community. This is a well known information theoretic measure that can be used to quantify the relationship between two signals. Given two signals with probability distributions P and Q, the Kullback Leibler divergence can be defined as:

$$D(P||Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \, dx \tag{1}$$

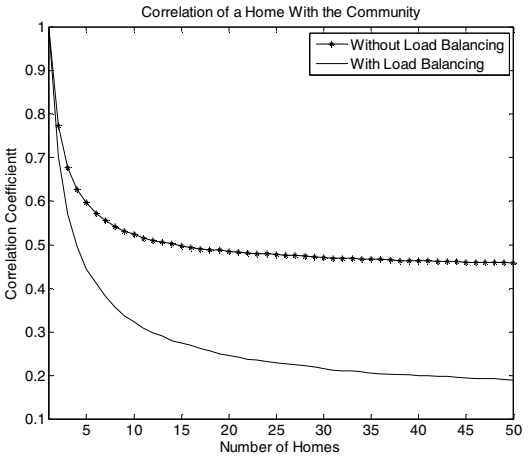


Fig. 7. Correlation between a home's demand and the community demand as produced by zone meters. Results presented are the average case of 100 iterations while increasing the number of homes in the community up to 50.

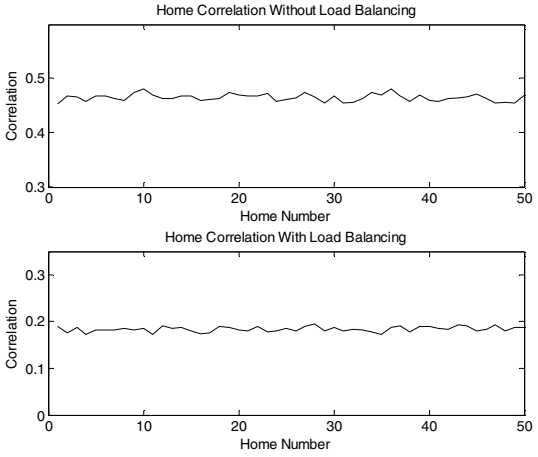


Fig. 8. Correlation of each simulated home with its community. All homes' signatures were distorted to a similar level. Results presented are the average case of 100 iterations for a community of 50 homes.

As shown in Fig. 9, the value of the Kullback Leibler divergence is zero for a single home, indicating identical distributions, and grows rapidly to saturate when accumulating about 10 homes. This indicates that accumulating a relatively small number of homes would have a good effect on masking individual homes' consumption profiles. Furthermore, as depicted in the figure, the use of load balancing helps achieve better privacy protection.

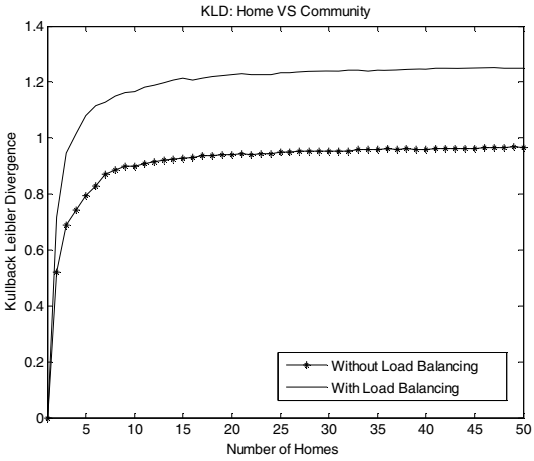


Fig. 9. KLD of a home's signature and the community demand as visible through zone meters. Results presented are the average case of 100 iterations while increasing the number of homes up to 50.

4.2 Appliance Detection

Assuming the availability of a library of appliance signatures [19], we assess the detection of the operation of appliances using cross correlation from readings produced by zone meters while increasing the number of homes in the community. Unlike previous simulations, the appliance to be detected is only run once by one member of the community. This ensures that a false detection is not caused by a duplicate signature of the appliance in question.

This case was simulated allowing the target appliance (i.e., the one to be detected) to be turned on randomly, following a uniform distribution throughout the 24 hour simulation interval with a step size of one minute; i.e., a total of 1440 possible time slots. We define a correct detection as one where the precise time slot was identified.

Fig. 10 shows the percentage of correct detections for a sample appliance as the number of homes in the community is increased. As the figure shows, the use of load balancing techniques causes a more rapid deterioration in detection accuracy, i.e., it achieves better privacy protection. Fig. 11 shows the error in detection, in time slots, as a function of the number of homes in the community. As shown, the detection becomes more distant from the real start time as the number of homes is increased. Furthermore, the use of load balancing causes an increase in the detection error, which implies better preservation of privacy.

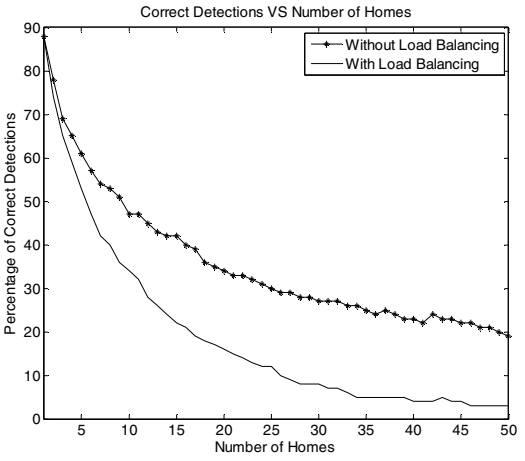


Fig. 10. Detection of a sample appliance using cross correlation deteriorates as the number of homes in the community is increased. Results presented are based on 100 detection attempts while increasing the number of homes up to 50.

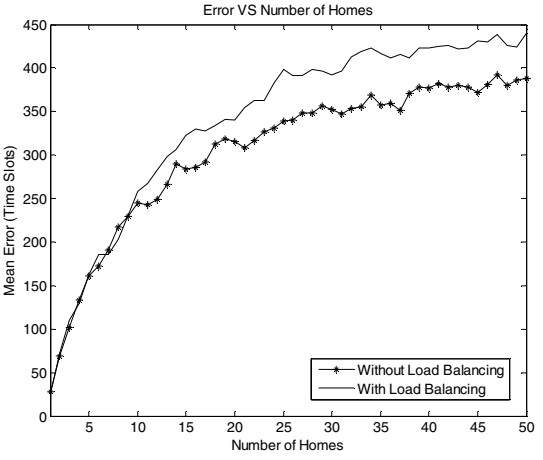


Fig. 11. The mean error in appliance signature detection increases as the number of homes is increased. Results presented are based on 100 detection attempts while increasing the number of homes up to 50.

5 Conclusion

In this paper, we showed that it is possible to achieve the objectives of smart metering without compromising the privacy of residential consumers. In our approach, home meters function as service providing modules rather than data acquisition devices allowing them to provide the desired functionality, such as time of use billing, while

eliminating the need to disclose users' high frequency readings to the utility company. Collecting high frequency readings is done at a higher level such as at the local step-down transformer which allows utility companies to achieve their operational objectives. While our approach requires home meters to be trusted, this is more feasible than attempting to secure the meters, the transmission network, and all the processing facilities. Furthermore, we also showed that our approach achieves better privacy protection when consumers opt to use load balancing techniques. Our results imply that, with the proposed approach, the more efficient operation of the grid can result in better privacy protection for individual customers.

References

1. The Ontario Smart Metering Initiative, http://www.consumerscouncil.com/site/consumers_council_of_canada/assets/pdf/SM_Report.pdf
2. Collier, S.E.: Ten steps to a smarter grid. In: Proc. IEEE Rural Electric Power Conference, REPC 2009, pp. B2–B7 (April 2009)
3. Caron, S., Kesidis, G.: Incentive-based energy consumption scheduling algorithms for the smart grid. In: Proc. First IEEE International Conference on Smart Grid Communications, SmartGridComm, pp. 391–396 (2010)
4. Chen, C., Kishore, S., Snyder, L.V.: An innovative RTP-based residential power scheduling scheme for smart grids. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 5956–5959 (2011)
5. Mohsenian-Rad, A.H., Wong, V., Jatskevich, J., Schober, R.: Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. In: Proc. IEEE PES Conference on Innovative Smart Grid Technologies, pp. 1–6 (2010)
6. Liang, J., Ng, S.K.K., Kendall, G., Cheng, J.W.M.: Load signature study—part i: basic concept, structure, and methodology. IEEE Transactions on Power Delivery 25, 551–560 (2010)
7. Liang, J., Ng, S.K.K., Kendall, G., Cheng, J.W.M.: Load signature study—part ii: disaggregation framework, simulation, and applications. IEEE Transactions on Power Delivery 25, 561–569 (2010)
8. Hart, G.W.: Nonintrusive appliance load monitoring. Proceedings of the IEEE 80(12), 1870–1891 (1992)
9. Quinn, E.: Privacy and the new energy infrastructure. Working Paper Series (2009), http://ssrn.com/abstract=1370731
10. Cavoukian, A.: Privacy by design: achieving the gold standard in data protection for the smart grid, http://www.ipc.on.ca/images/resources/achieve-goldstnd.pdf
11. Cavoukian, A.: Smart privacy for the smart grid: embedding privacy into the design of electricity conservation, http://www.ipc.on.ca/images/resources/pbd-smartpriv-smartgrid.pdf
12. Cavoukian, A.: Operationalizing privacy by design: the Ontario smart grid case study, http://www.ipc.on.ca/images/Resources/pbd-ont-smartgrid-casestudy.pdf

13. Kalogridis, G., Efthymiou, C., Denic, S.Z., Lewis, T.A., Cepeda, R.: Privacy for smart meters: towards undetectable appliance load signatures. In: Proc. IEEE Smart Grid Commun. Conf., Gaithersburg, Maryland, pp. 232–237 (2010)
14. Kalogridis, G., Zhong, F., Basutkar, S.: Affordable privacy for home smart meters. In: Proc. IEEE Int. Workshop Smart Grid Security Commun., SGSC, Busan, Korea, May 26-28, pp. 77–84 (2011)
15. Efthymiou, C., Kalogridis, G.: Smart grid privacy via anonymization of smart metering data. In: Proc. IEEE Smart Grid Commun. Conf., Gaithersburg, Maryland, pp. 238–243 (2010)
16. Tomosada, M., Sinohara, Y.: Virtual energy demand data: estimating energy load and protecting consumers' privacy. In: Proc. 2011 IEEE PES Innovative Smart Grid Technologies, ISGT 2011, Medellin, Colombia, pp. 1–8 (2011)
17. Mitchell, C. (ed.): Trusted computing. Institution of Electrical Engineers (2005)
18. Pearson, S.: Trusted computing platforms, the next security solution. HP Labs (2002)
19. Richardson, I., Thomson, M., Infield, D.: A high-resolution domestic building occupancy model for energy demand simulations. Energy and Buildings 40(8), 1560–1566 (2008)
20. Ontario Demand and Market Prices. The Independent Electricity System Operator, http://www.ieso.ca/

Developing an Intelligent Intrusion Detection and Prevention System against Web Application Malware

Ammar Alazab[1], Michael Hobbs[1], Jemal Abawajy[1], and Ansam Khraisat[2]

[1] School of Information Technology, Deakin University, Waurn Ponds, Australia
{aalazab,mick,jemal.abawajy}@deakin.edu.au
[2] University of Ballarat, Ballarat, Australia

Abstract. Malware authors are continuously developing crime toolkits. This has led to the situation of zero-day attacks, where malware harm computer systems despite the protection from existing Intrusion Detection Systems (IDSs). We propose an Intelligent Intrusion Detection and Prevention System (IIDPS) approach that combines the Signature based Intrusion Detection system (SIDS), Anomaly based Intrusion Detection System (AIDS) and Response Intrusion Detection System (RIDS). We used a risk assessment approach to determine an appropriate response action against each attack event. We also demonstrated the IIDPS make the detection and prevention of malware more effective.

Keywords: Intrusion Detection System, Response Action, Malware, Signature Base Detection, Anomaly Base Detection, Web application.

1 Introduction

Malicious software (Malware) in web applications may result in stealing of confidential data, breaking of data integrity, reduced availability, causing damage or risk of data loss. Thus malware prevention and detection is vital to secure the web applications[1]. Recent trends in web application malware have become a major threat and they are increasing in complexity and evolving rapidly as systems provide more opportunities for more automated activities[2]. Furthermore, the damage caused by web application malware to individuals and businesses have dramatically increased. Today, writers of malware develop sophisticated techniques for concealing or constantly changing their attacks to evade established detection software. Attackers, who achieve unauthorized access to financial web applications, are causing losses to the financial sectors and there is no one single technique that can stop them[3]. Generally, the attacker developed new sophisticated techniques to specifically target and compromises web applications. As a result, attackers have access to other user's data. Furthermore, every year the quantity and creativity of web application hacks growths and the threat impact from these attacks increases rapidly, costing organizations millions every year[4].Moreover, the new generation of cybercrime is high degree of stealthiest and the attacker developing tool kits attacks that pose severe challenges to protect internet users. These crime tool kits such as Zeus and SpyEye, which have powerful capability

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 177–184, 2013.

of attacks and have led to the threat of zero-day attacks, have showed a necessity to identify an Intelligent Malware Detection and Prevention System.

However, most exiting intrusion detection systems suffer from critical problems, such as: low detection accuracy; high false alarm rate; and the difficulties in dealing with the new attack. In this paper, we propose IIDPS for the efficient prevention and detection of malware.

Protecting web application from malicious attacks is an essential issue. Within this, intrusion prevention and intrusion detection systems have been the topic of a lot of research and have been suggested in a number of papers[5][6][7]. Nevertheless, the action that should follow the functionality of prevention and detection, namely response action, has needed to be involved as a primary function against any potential attack.

This paper is organized as follows. In Section 2, we present the background and related work. In Section 3, we describe the design and structure of our IIDPS model. Section 4 provides the conclusion to this paper.

2 Background and Related Work

Initially intrusion detection techniques mostly relied on matching to signatures patterns of well-known malware for triggering a detection decision. This style of detection strategy is usually known as Signature Base Detection (SBD). Nevertheless, it is very hard for SBD to detect zero-day attack, since such a malware example would have previously unidentified signatures. Thus anomaly base detection has attracted many researchers to overcome for this problem. Unfortunately anomaly detection systems suffer from high false negative, that is, the incorrect classification of valid software as malware[1].

Based on the input data sources the IDSs are examine, there are two main types of IDSs: network-based IDSs and host-based IDSs. Host-based IDSs (HIDS) examine host-bound audit sources such as application system audit, operating system, system logs, or database logs. A HIDS detector play significant role for detection inside attacks that do not involve network traffic. While network-based IDSs (NIDS) examine network packets that are taken from a network. Network-based IDS can be implementing to protect several hosts that are connected to a network. NIDS can report an attack that could be launched from the external at an earlier stage, before the attacks actually reach the host. However, NIDSs have the capacity problem to examine all packets in a high speed network.

Based on examination techniques, there are two approaches to analysing events using IDSs intrusion detection techniques can be categorized into two classes: signature based detection and anomaly based detection.

Up until now, there have only been a few approaches that have implemented IDS to find anomalies in web applications using a (SIDS) and an (AIDS). However, very few have used a combination of the two approaches[6]. Unfortunately none of them can guarantee a high level of security on web applications due to the web application architecture. Regarding current research for intrusion detection on web applications, Table 1 provides a summary of the research in developing an Intrusion detection system.

Table 1. Current Research in the Area of Anomaly Based Detection on Web Application

Name	Comments	Detection	Prevention	Response
(Vigna et al., 2009) [7] Reducing errors in the anomaly-based detection of web-based attacks through the combined analysis of web requests and SQL queries.	Reduce FP and FN but doesn't validate with Data Mining Algorithms.	NO	YES	NO
(Maggi, Robertson, Kruegel, & Vigna, 2009) [8] Protecting a Moving Target: Addressing Web Application Concept Drift	Anomaly- based detection of changes in web application	YES	NO	NO
(Kruegel, 2008) [9] Anomaly Detection of Web-based Attacks.	Anomaly detection with parameter profiles associated web applications (length and structure of parameters) from the analyzed data.	YES	NO	NO
(W. K. Robertson, 2010) [10] Detecting and preventing attacks against web applications.	Detection system that accurately detects attacks against web applications.	YES	NO	NO
(Cova, Balzarotti, Felmetsger, & Vigna, 2007) [11] Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications	Anomaly detection by Learning the relationships between the application's execution and the application's internal.	YES	NO	NO
(Dagorn, 2008)[12] WebIDS: A Cooperative Bayesian Anomaly-Based Intrusion Detection System for Web Applications.	Learning-based anomaly detection system for Web applications	YES	NO	NO

3 Intelligent Intrusion Detection and Prevention System (IIDPS)

As shown in Section 2, traditional IDS have restrictions, including low flexibility, inability to distinguish novel attacks, high cost, slow updates and lacks extensibility. It also shows that both SIDS and AIDS have drawbacks such as low detection

zero-day attack. The aim of the new approach is design and develops an effective IIDPS that address the weakness of SIDS and AIDS. Our IIDPS combines SIDS, AIDS and RIDS to become an IIDPS. Figure 1 shows an overview of the proposed Intelligent Intrusion Detection and Prevention System. In our system, AIDS help to detect unknown attacks, while SIDS detects known attacks. The basic idea of the new system is to take benefits from both SIDS and AIDS to create effective IDS. The IIDPS has three stages; the SIDS stage, the AIDS stage and the response action stage as shown in Figure 1.

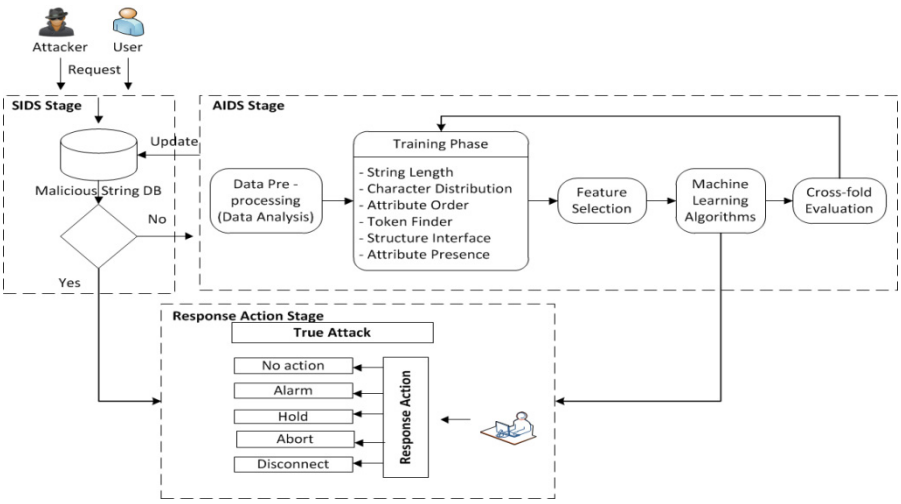


Fig. 1. Overview of IIDPS and Response Action Model

3.1 SIDS Stage

The SIDS stage simply uses pattern matching to handle the received request from clients. Whether or not this request is legitimate or illegitimate, SIDS will detect a known malicious string attack that has been previously stored in the SIDS signature database. If the request is passed to the final stage has the same pattern as found on the malicious string database, it means the request will be identified on the system as a true attack. As a consequence, response action is taking the appropriate response action. Otherwise, if the received request is not found in the malicious string, the AIDS stage handles the request.

3.2 AIDS Stage

The second stage is the AIDS stage. The main idea of this stage is to overcome the shortfalls of the SIDS stage. The main assumption is that any request received from users is an anomaly request, unless proven otherwise. In the AIDS stage, the system builds the profile of users by using data that is accepted as normal behaviour. Then it monitors the activities of new users and compares the new data with the obtained profile and tries to detect deviations.

In this stage, the system gathers information from the user request such as request length, character distribution and particular token, and if any suspicious event is detected, the system will store it in the signature database. The reason from the store signature in the database is taking precautions against the new attack in future requests.

Once the users profile has been built, the system can decide if the user activity is a normal or abnormal behaviour. The profile information collected from the users' activities by using the learning mode enables identification of the appropriate response to any attack, as shown at the bottom of Figure 1.

3.3 Response Action Stage

The final element of our approach is taking appropriate response actions against a request if it is found to be an anomaly. A response action is a set of instructions that is carried out for a given attack. Response actions are triggered by the response action policy in reply to an attack which is detected by SIDS or AIDS. Once SIDS and AIDS detect an attack then this stage comes to reacts with attacks. The Response Action has two stages, risk assessment stage and response reaction stage.

Risk Assessment

The main purpose for risk assessment is to estimate the risk level of an attack. We have adopted the DREAD model from Microsoft to help calculate risk and rate the threats[13]. By using the DREAD model, it is possible to arrive at the risk rating for a given threat by asking the following questions:

- **D**amage potential: How great is the damage if the vulnerability is exploited?
- **R**eproducibility: How easy is it to reproduce the attack?
- **E**xploitability: How easy is it to launch an attack?
- **A**ffected users: As a rough percentage, how many users are affected?
- **D**iscoverability: How easy is it to find the vulnerability?

We applied DREAD risk assessment to identify the risk level for an application attack. We use the risk matrix to determine the risk assessment process. These matrices provide a qualitative risk ranking that classifies the degree from very high to very low as shown in Table 2. The probability of risk ranges from zero to one. The threat impact can be classified in five states as shown in the following sets:

Probability of Risk= {Very Low, Low, Medium, High, Very High}

Impact of Risk= {Very Low, Low, Medium, High, Very High}

Table 2. Standard terms for severity quantification

Probability	Description
Very high	Expected to occur with almost certainty
High	Expected to occur
Medium	Likely occur
Low	Very unlikely to occur
Very low	Almost no possibility of occurring

We calculate each query risk and evaluate the probability of the risk occurring against the security impact using the following equations

Risk =Probability x Impact

Expected Value= [Risk Probability Value]* [Risk Impact Value]

On the vertical axis, there is probability of the risk occurring, thus a higher chance of that risk occurring and becoming an issue. The horizontal axis shows the level of impact in the assumption that the risk will occur. As shown in Figure 2, the value outputs near to zero point to normal features, while outputs near to one indicate anomalous ones.

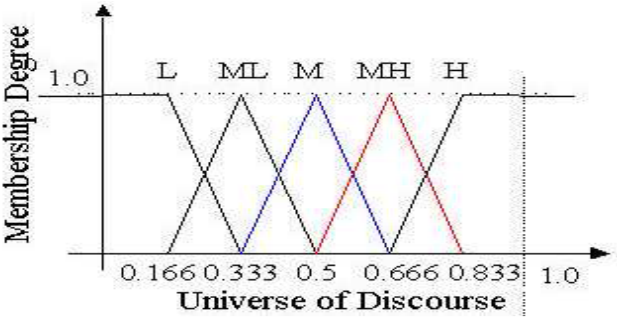


Fig. 2. Rule Action

One of the advantages of this approach is being able to show risks and identify how risky they are on the database. If all the risks are clustered in the top right of the diagram, then evidently the database is very risky. In other words, it may be exploited by a malicious writer.

Response Reaction

In this stage, we will identify the best response against a database threat according to the level of severity. Once the risk is estimated from the previous stage, our approach can and determines appropriate action response. The reaction of our approach is responsible for providing a corresponding response action when an anomaly activity is detected.

Once the users' try request a malicious string, the response action will be executed. This response action will handle this request according to severity methods as shown in Table 3. There are six principle methods to handle risk. Table 3 shows each response action according to severity request.

Table 3. Response Actions

Severity Level	Severity Level	Description
Low Severity	No action	The system will process as normal
	Alarm	Sent notification to DBA
Medium Severity	Audit	The request is audit
	Hold	The user requests is aborted
High Severity	Disconnect	The user session is disconnect
	Refuse	The user request is refused

Once the risk has been calculated, an appropriate action will be executed according to the severity level. For example if damage potential is very high; reproducibility is very high; exploitability is very high ; affected users is high, and discoverability is very high then risk level is consider very high.

4 Conclusion

A detection system with a response system would be highly reliable against suspicious behaviour, as it is able to detect and react to maliciously requests that could possibly be a zero day attack. In this paper, we developed IIDPS with a response action that provides an early stage detection system. Our IIDPS combines SIDS, AIDS and RIDS to become an IIDPS. Our approach of IIDPS is capable of preventing and distinguishing various types of abnormal activity. SIDS was used to recognize known attacks, while AIDS was used to recognize unidentified attacks. A risk assessment was also done in order to respond to the attack, with several response techniques used to minimize the damage caused by malicious activities.

References

1. Alazab, A., Abawajy, J., Hobbs, M.: Web Malware That Target Web Application. In: Caviglione, L., Coccoli, M., Merlo, A. (eds.) Social Network Engineering for Secure Web Data and Services. IGI Global, USA (2013)

2. Alazab, A., Alazab, M., Abawajy, J., Hobbs, M.: Web Application Protection against SQL injection Attack. In: Proceedings of the 7th International Conference on Information Technology and Applications, pp. 1–7. IEEE (2011)
3. Alazab, M., Ventatraman, S., Watters, P., Alazab, M., Alazab, A.: Cybercrime: The Case of Obuscated Malware. In: 7th International Conference on Global Security, Safety & Sustainability (2011)
4. Alazab, M., Venkatraman, S., Watters, P., Alazab, M.: Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures. In: Australasian Data Mining Conference (AusDM 2011), pp. 171–182. ACS (2011)
5. Shameli-Sendi, A., Ezzati-Jivan, N., Jabbarifar, M., Dagenais, M.: Intrusion response systems: survey and taxonomy. Int. J. Comput. Sci. Network Secur (IJCSNS) 12(1), 1–14 (2012)
6. Alazab, A., Hobbs, M., Abawajy, J., Alazab, M.: Using feature selection for intrusion detection system. In: International Symposium on Communications and Information Technologies (ISCIT), pp. 296–301. IEEE (2012)
7. Vigna, G., Valeur, F., Balzarotti, D., Robertson, W., Kruegel, C., Kirda, E.: Reducing errors in the anomaly-based detection of web-based attacks through the combined analysis of web requests and SQL queries. Journal of Computer Security 17, 305–329 (2009)
8. Robertson, W., Maggi, F., Kruegel, C., Vigna, G.: Effective anomaly detection with scarce training data. In: Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA (2010)
9. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 251–261. ACM (2003)
10. Robertson, W.K., Adviser-Kemmerer, R.A., Adviser-Vigna, G.: Detecting and preventing attacks against web applications. University of California at Santa Barbara (2009)
11. Cova, M., Balzarotti, D., Felmetsger, V., Vigna, G.: Swaddler: An approach for the anomaly-based detection of state violations in web applications. In: Kruegel, C., Lippmann, R., Clark, A. (eds.) RAID 2007. LNCS, vol. 4637, pp. 63–86. Springer, Heidelberg (2007)
12. Dagorn, N.: WebIDS: A Cooperative Bayesian Anomaly-Based Intrusion Detection System for Web Applications (Extended Abstract). In: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) RAID 2008. LNCS, vol. 5230, pp. 392–393. Springer, Heidelberg (2008)
13. `http://msdn.microsoft.com/en-us/library/ff648644.aspx`

Vulnerability Scanners Capabilities for Detecting Windows Missed Patches: Comparative Study

Mohamed Alfateh Badawy, Nawal El-Fishawy, and Osama Elshakankiry

Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, Egypt
mohamed.alfateh@owasp.org, {nelfishawy,osama1975}@hotmail.com

Abstract. Vulnerability scanners are automated tools that define, identify, and classify security holes (vulnerabilities) in a computer, server, network, or communications infrastructure. Scanners discover missed patches on target systems and report related vulnerabilities. Many of the current information security systems use vulnerability scanners as the main part in the risk assessment process. Others depend on the scanners output in the systems patch management. This paper assesses the effectiveness of depending on vulnerability scanners in the information security management system. It compares between four of the leading vulnerability scanners in the market and carries out a study of their effectiveness in detecting missed patches.

The results show the severity of relying on vulnerability scanners to discover system patches status. A number of false positive and false negative detections for the system patches are reported by each of the tested scanners. The severe level for some of the unreported missed patches ranked as critical that puts the system in a high risk and makes it vulnerable for different attacks.

Keywords: Vulnerability scanner, patch management, risk assessment.

1 Introduction

The increasing volume of attacks on the Internet has increased the demand for sophisticated tools and techniques to detect systems vulnerabilities and to perform vulnerability analysis. Minimizing this threat requires organizations to configure systems properly, use the latest software, and install the recommended security updates. Creating and communicating a documented security release and update policy is a vital part of any companys risk-management process [1].

Vulnerability scanning plays a main role to identify systems vulnerabilities during the vulnerability management process. Vulnerability scanners are automated tools that are used to perform system discovery, identify open ports and running services on the discovered system, and then analyze them for potential vulnerabilities. In addition, scanners can help in identifying outdated software versions, missing patches, and misconfigurations [2].

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 185–195, 2013.

On the other hand the system patches correct security and functionality problems in software and firmware. From a security perspective, security patches are most often of interest because they are mitigating software flaw vulnerabilities; i.e., applying patches to eliminate these vulnerabilities significantly reduces the opportunities for exploitation.

In enterprise networks, deploying software patches is not an easy task; the deployment should be managed through patch management process. Patch management is the process of identifying, acquiring, installing, and verifying patches for products and systems [3].

Agentless scanning using network scanners is one of three techniques introduced by National Institute of Standards and Technology (NIST) to perform patch management in enterprise networks [4]. Moreover, research was introduced enhancing the intrusion detection system based on the vulnerability scanners detections [5].

In this paper, a comparative study is performed to determine to what extent vulnerability scanners could be trusted in detecting missed patches or even in verifying the installation of a specific patch.

Although Gartner generates an annual report to compare vulnerability scanners [6][7], the evaluation criteria are mainly based on the vendors market share and the scanners add-ons features such as integration with other security products, reporting capabilities, deployment options, management features and compliance check.

This paper focuses on the scanners capabilities of detecting Microsoft windows missed patches. In addition, it shows the severity level for the unreported patches. It starts by determining and studying all related patches released from Microsoft. Next, in different system conditions, it performs the scans using each vendor separately. Finally, it compares the scan results with Microsoft windows updates.

The rest of the paper is organised as following. Section 2 describes the vulnerability assessment and vulnerability scanners. Section 3 describes Microsoft update, Section 4 illustrates common security impacts related to Microsoft missed patches. Section 5 presents the experimental results and, finally, Section 5 concludes the paper.

2 Vulnerability Assessment and Vulnerability Scanners

Vulnerabilities are weaknesses in software that may enable an attacker to compromise the integrity, availability, or confidentiality of that software. When the term vulnerability assessment is used in the context of vulnerability scanners it means the process of finding known vulnerabilities in a network [8]. This process identifies vulnerabilities so they can be eliminated before exploited by malicious software or hackers. The vulnerabilities that constitute threats in a network include software defects, unnecessary services, misconfigurations and unsecured accounts.

During network systems lifetime the security must be constantly updated and developed to encounter new and enhanced vulnerabilities. NIST has described

a model for security maintenance. The model recommends using vulnerability scanners among other tools, in regular testing to make sure that the network is secured [9].

A vulnerability scanner starts like a port scanner and tries to identify all the hosts running in the defined IP range. When the hosts have been found, the scanner tries to find all opened ports and corresponding services on all active hosts, and then identifies vulnerabilities in the scanned host. This is done through comparing running operating systems and software applications with known vulnerabilities stored in a database [9]. In some cases, vulnerability is identified from the information of a banner and version test. In other cases, the scanner makes a complete exploitation of the vulnerability to insure its existence.

2.1 Vulnerability Scanners False Alerting

A false-positive is when the vulnerability scanner reports an error that is not present. On the other hand, the false-negative is when the vulnerability scanner missed reporting an existing vulnerability.

There are a number of reasons of why a false alerting occurs. The false alerting may happen because of the technique used to check for vulnerabilities. Some scanners just look for signs such as registry entries in Microsoft Windows operating systems to identify that a specific security patch or update has been implemented. Other scanners look at relevant DLL and other files affected by applying the patch or update. While the latter is slightly slower, it is more accurate and reliable [10].

There are many instances where a Windows operating system can have a security patch seemingly applied, but not actually in effect. For example, there could have been an error during the patch update process or the patch required a reboot to take effect. Moreover, the network connection between the scanner and the vulnerable system might drop some scanning packets and affect the scanner detection. Also, the vulnerable service running on the target system might have become temporarily unavailable during the scanning requests.

Another very important cause of false alerting is the time between when vulnerability is disclosed and when a scanner database is updated. It is very common that vulnerabilities can get reported where the scanner is delayed in updating the scanning database to include checks for those vulnerabilities.

2.2 Tested Vulnerability Scanners

To perform our comparative experiments, we used four of the leader vulnerability scanners on the market. In addition and after each scan, we used Microsoft Baseline Security Analyzer (MBSA) [11] to verify the scanning output. The following is a brief description for the scanners we used.

McAfee Vulnerability Manager. Was formerly known as Foundstone [12]. MVM can be integrated with other McAfee products as well as a large number of third-party security products. In 2011 and 2012, MVM was rated as a strong positive in Gartner report for the vulnerability assessment.

Retina Network Security Scanner. Retina scanner was developed by eEye Digital Security and has been acquired by BeyondTrust in May 2012 [13]. Retina offer a Fix-it feature to automatically correct some system security issues discovered during the scanning including registry settings and file permissions. Retina rated positive in Gartner report.

Nexpose Vulnerability Management from Rapid7. In 2009 Rapid7 acquired the open-source Metasploit framework penetration testing engine, and released a commercial version of it in 2010. Nexpose integrates with Metasploit to validate security risks for the discovered vulnerabilities [14]. The scanner was rated strong positive in Gartner reports.

Nessus Vulnerability Scanner from Tenable. The "Nessus" project was started by Renaud Deraison in 1998 to provide a free remote security scanner to the Internet community. In 2005, Tenable changed Nessus to a proprietary (closed source) license [15]. In 2012 the scanner rated strong positive in Gartner report.

3 Microsoft Update

Microsoft update is a service from Microsoft that provides a listing of Microsoft software updates, drivers, and hotfixes. Microsoft offers important, recommended, and optional updates.

Important updates provide significant benefits such as improved security and reliability. Recommended updates are those enhance the performance and the computing OS experience. Optional updates might include new or updated driver software for a specific device.

A security update is a widely released fix for a product-specific, security-related vulnerability. Microsoft security updates are accompanied by two documents: a security bulletin and a Microsoft knowledge base article. Microsoft schedules the release of the security update and the security bulletin on the second Tuesday of the month at 10:00 AM in the Pacific Time zone. The security bulletin advance notification occurs three business days before this [1].

Microsoft also provides service pack (SP) update. PS is a tested, cumulative set of all hotfixes, security updates, and critical updates.

A single security update often addresses multiple vulnerabilities from the Common Vulnerabilities and Exposures (CVE) database each of which is listed in a corresponding Microsoft security bulletin along with any other relevant issues.

Security vulnerabilities are rated based on their severity. The severity rating is indicated in the Microsoft security bulletin as critical, important, moderate, or low. Microsoft evaluates each issue and quantifies an issues impact objectively on a technical level for default configurations. Based on this analysis and the maximum security impact, Microsoft supplies a rating in the security bulletin. Table1 defines the four Microsoft severity ratings and their corresponding impact [1].

Table 1. Microsoft Severity Ratings

Rating	Definition
Critical	A vulnerability whose exploitation could enable the propagation of an Internet worm with little or no user action.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users data, or of the integrity or availability of processing resources.
Moderate	A vulnerability whose exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
Low	A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

4 Security Impacts Related to Microsoft Released Patches

Comparing operating system vulnerabilities to non-operating system vulnerabilities require determining whether a particular program or component should be considered part of an operating system. Microsoft update service releases security updates for different Microsoft product like Internet Explorer and Microsoft Office as well as the security updates released for the operating system components.

Microsoft provides information about the availability of Proof-of-Concept (PoC) exploit code or active attacks related to vulnerabilities addressed by Microsoft security updates [1]. The maximum security impact for each released update is mentioned in the Microsoft Security Bulletin. The following are the common Security Impact reported in the Microsoft security bulletin:

- Remote Code Execution
- Denial of Service
- Information Disclosure
- Elevation of Privilege
- Tampering
- Spoofing

5 Comparative Study

In our experiments, we used the latest available version for the tested scanners with a license for each scanner that gave us an access to the latest released scanner database on-time updates. Also, Windows server 2008 was used as the target machine. Table 2 shows the versions for the used scanners.

Table 2. Tested Vulnerability Scanners' Version

Scanner Name	Version
MVM	7.5
Retina	5.18
Nessus	5.0.2
Nexpose	5.5.12

In order to detect the effects of installing SP on the accuracy of the scanning, the test was conducted through two phases. At the first phase, the scans were performed against the target server before installing any updates or SPs. At the second phase the scans was repeated after installing the SP2.

We start our experiments by listing all released Microsoft security updates for windows server 2008 and understand the new patches those replaced older ones.

5.1 First Phase

In this phase, the scans were performed before installing any updates or SP. To verify the scanning output and to detect how often the scanners update their scanning data-base, the scans were performed two times before the Microsoft Tuesday updates and two times after the Microsoft Tuesday updates respectively. In the latter, the first scan was performed one day after disclosing Microsoft updates and the second scan was performed one week later.

By running the scans before Microsoft Tuesday updates, the two results were the same for each scanner. Fig. 1 shows the comparative analysis for the scanners reporting for both false positive and false negative.

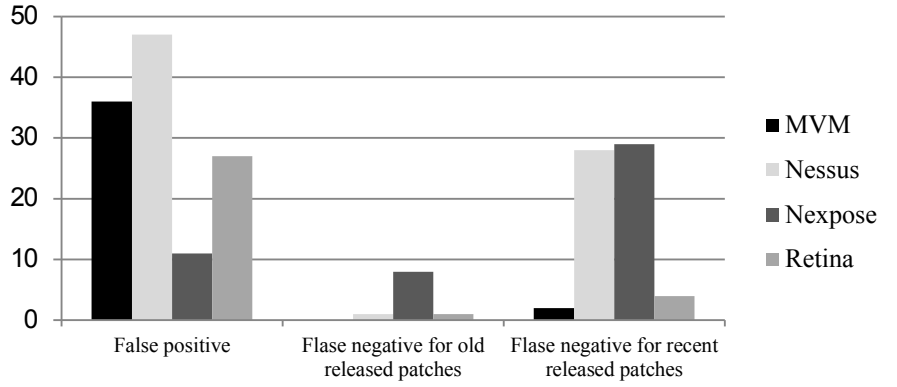


Fig. 1. Vulnerability scanners false positive and negative results before Windows Tuesday update

It is noticed that all the scanners have a higher detection ratio for the old Microsoft patches than the recent released patches. In addition, MVM and Retina were found the only scanners that significantly mentioned Microsoft replacement patches. Nessus has a higher number of false positive detections because of reporting many missed Microsoft patches that were replaced with newer ones.

After Microsoft released the Tuesday updates, the scanners were run two times. (Fig. 2 shows the output results of running the scanners one day after the release of Microsoft updates).

The results show that all scanners except Nexpose had updated their scanning data-base and reported the new released windows updates. The released windows updates include some patches that replaced many of the old Microsoft patches. Also, it is shown that none of the scanner reported the replaced patches, which explains the increase in the false positive.

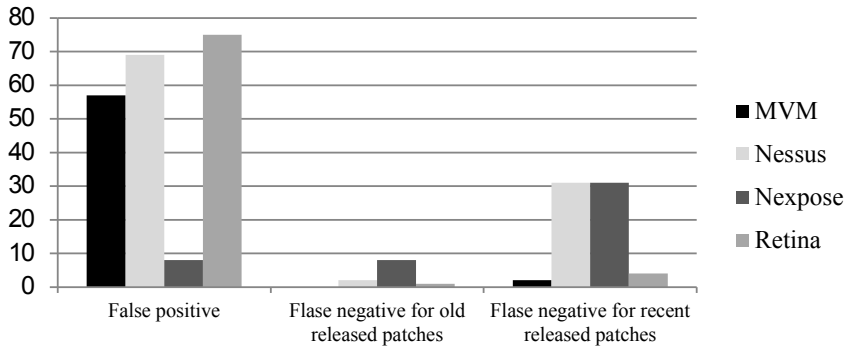


Fig. 2. Vulnerability scanners false positive and negative results before Windows Tuesday update

By performing the scans one week after the patches release date; the only noticeable result was the significant decrease of the false positive reported by MVM scanner. This is because MVM was the only scanner that updates the replaced patches in its reporting; Fig. 3 shows the comparative results for all scanners.

The results of the scans performed in the first phase show the significant effect of misreporting the replaced Microsoft patches, Moreover; the results show some limita-tion for all the scanners in detecting the recent released Microsoft updates rather than the old ones

Table 3 shows the attacks related to the missed patches reported for each of the tested scanners:

Table 4 shows the average severe level for the missed patches reported for the tested scanners

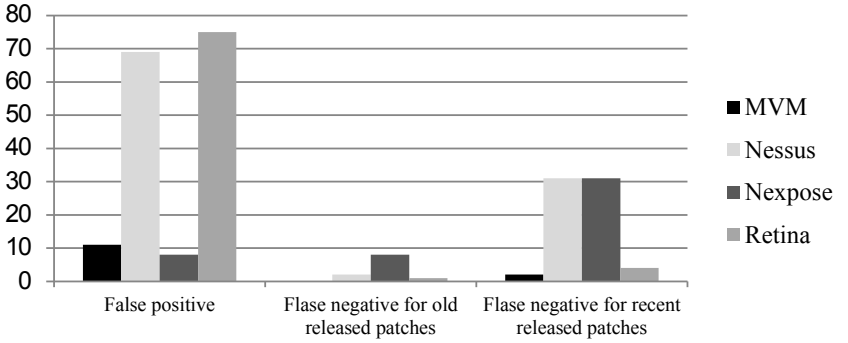


Fig. 3. Vulnerability scanners false positive and negative results before Windows Tuesday update

Table 3. Unreported Patches Related Attacks

	MVM	Nessus	Nexpose	Retina
Remote Code Execution	2	23	25	4
Denial of Service	0	2	2	0
Information Disclosure	0	2	3	1
Elevation of Privilege	0	5	4	0
Tampering	0	0	1	0
spoofing	0	0	1	0
Security Feature Bypass	0	2	2	0

Table 4. Scanners Unreported Patches Severe Level

	MVM	Nessus	Nexpose	Retina
Critical	2	11	12	1
Important	0	15	11	2
Moderate	0	6	8	2
Information	0	0	2	0

5.2 Second Phase

In this phase, the scanners were used to scan the same machine after installing SP2, and the latest dotNet framework version. Again, the scans were performed twice; before Microsoft released the Tuesday update and after.

The comparative result for the scanners before Microsoft Tuesday patch updates is shown in Fig. 4.

The results show that after installing the SP2 all scanners were able to detect most of the missed patches. The increase of the false positive for MVM scanner was due to incorrect reporting for some replaced patches. In addition MVM was the only scanner that exactly reported all the replaced patches, Retina reported some of them, and both Nessus and Nexpose did not report any.

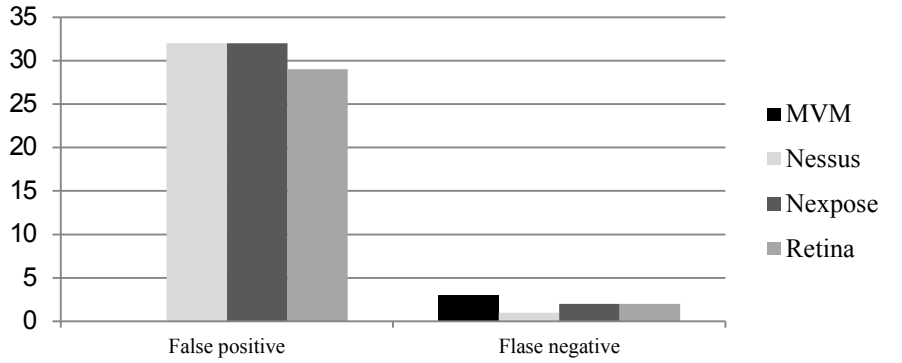


Fig. 4. Vulnerability scanners false positive and negative results before Windows Tuesday update

The scans were repeated one day after the releasing of the new Microsoft Tuesday patches, Fig. 5 shows the results for each scanner.

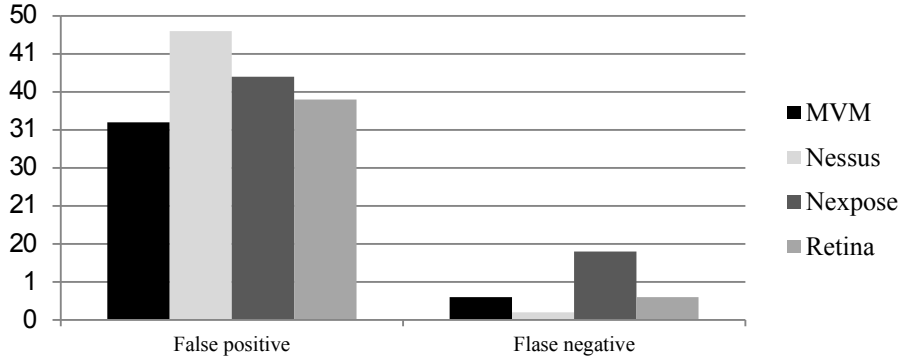


Fig. 5. Vulnerability scanners false positive and negative results before Windows Tuesday update

The results show that all scanners except Nexpose updated their scanning database and reported the new released patches; however, the increase of the false positive for MVM scanner were due to that some new release patches replaced old ones and MVM did not update the replaced patches.

Table 5 shows the attacks related to the missed patches reported for each of the tested scanners.

Table 6 shows the average severe level for the missed patches reported for the tested scanners.

Table 5. Second Phase Unreported Patches Related Attacks

	MVM	Nessus	Nexpose	Retina
Remote Code Execution	3	1	5	2
Denial of Service	0	0	1	0
Elevation of Privilege	0	0	3	1

Table 6. Second Phase Scanners Unreported Patches Severe Level

	MVM	Nessus	Nexpose	Retina
Critical	3	1	3	1
Important	0	0	5	1
Moderate	0	0	1	1

Finally, the scans were performed after installing the all required patches on the machine. There was only one false positive reported by Retina scanner; i.e., the rest of the scanners did not report any.

6 Conclusions

Vulnerability assessment is an important mechanism to provide assurance of appropriate level of confidentiality, integrity and availability of information. It could be used in identifying potential security exposures. Vulnerability scanners are the handy tools used to discover system vulnerabilities in the assessment process. This paper describes a comparative study to find out to what extent a vulnerability scanner can be used to secure a network. This paper compares between four of the leading vulnerability scanners in the market to find out to what extent a vulnerability scanner could be used to secure a network; to find out the scanners effectiveness in detecting Microsoft windows missed patches.

The analysis of the scanners results shows that scanners do not only report unneeded system patches, but also they miss a number of severe patches. After installing the service pack some scanners detection was improved. One of the main challenges for the vulnerability scanners is updating their database with the newly released patches and the replacement for old ones. The analysis of the scanners output shows that not all scanners take into accounts the replaced patches. Moreover, some scanners misreporting some of the replaced patches, consequently increasing the number of false detection.

The findings in this paper points out that system administrators should not depend only on the vulnerability scanners to check for system missed patches, or in tuning other security controls like intrusion detection systems.

References

1. Microsoft Corporation: Microsoft Security Update Guide. 2nd edn. (June 2011)
2. Nist, Aroms, E.: NIST Special Publication 800-115 Technical Guide to Information Security Testing and Assessment. CreateSpace, Paramount, CA (2012)
3. Danforth, M.: Scalable patch management using evolutionary analysis of attack graphs. In: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, ICMLA 2008, pp. 300–307. IEEE Computer Society, Washington, DC (2008)
4. Souppaya, M., Scarfone, K.: Guide to enterprise patch management technologies. National Institute of Standards and Technology, NIST SP 800-40 (September 2012)
5. Yang, G., Chen, D., Xu, J., Zhu, Z.: Research of intrusion detection system based on vulnerability scanner. In: 2010 2nd International Conference on Advanced Computer Control, ICACC, pp. 173–176 (2010)
6. Kavanagh, K.: Marketscope for vulnerability assessment. Gartner, Inc. (August 2011) G00230435
7. Kavanagh, K., Nicolett, M.: Marketscope for vulnerability assessment. Gartner, Inc. (April 2011) G00211846
8. Nilsson, J., Virta, V.: Vulnerability scanners. Royal Institute of Technology, Stockholm (2006)
9. Wack, J., Tracy, M., Souppaya, M.: Guideline on network security testing. National Institute of Standards and Technology, NIST SP 800-42 (October 2003)
10. Beale, J., Deraison, R., Meer, H., Temmingh, R., Walt, C.V.D.: Nessus Network Auditing. Syngress Publishing (2004)
11. Microsoft: Microsoft baseline security analyzer v2.2 (July 2010), `http://microsoft.com/en-us/download/details.aspx?id=7558`
12. McAfee: Mcafee vulnerability manager v7.5 (December 2012), `http://www.mcafee.com`
13. BeyondTrust: Retina network security scanner v5.18 (2012), `http://beyondtrust.com`
14. Rapid7: Nexpose vulnerability management v5.5.12 (2012), `http://www.rapid7.com`
15. Tenable: Nessus vulnerability scanner v5.0.2 (February 2012), `http://www.tenable.com`

Elderly Healthcare Data Protection Application for Ambient Assisted Living

Qing Tan[1], Nashwa El-Bendary[1,2], Frédérique C. Pivot[1], and Anthony Lam[3]

[1] Athabasca University, Alberta, Canada
{qingt,nashwaelbendary,fpivot}@athabascau.ca
[2] Arab Academy for Science,Technology, and Maritime Transport, Cairo, Egypt
nashwa.elbendary@aast.edu
[3] Edmonton Chinatown Care Centre, Edmonton, Alberta, Canada
alam@edmccc.net

Abstract. The increasing aging of the population requires new kinds of social and medical intervention and the availability of different supportive services for elderly people. Falling is one of the events that usually occur to elderly persons with resultant morbidity ranging from soft injury through to fractures and possibly death. Due the fragility of the elderly persons, falls should be avoided at all costs. New applications and services have been developed allowing the elderly people to be continuously monitored, however an adequate response to the needs of the users will imply a high percentage of use for personal data and information. This article introduces the data protection rules that have to be considered in elderly healthcare facilities for protecting residents' privacy and sensible data that is being shared between persons and applications. Also, this article proposes an automatic video surveillance system for elderly protection via falls detection and prevention. The proposed system utilized the integration of motion detection sensors embedded in network cameras with smart mobile phones in order to develop a platform for pervasive fall detection and prevention. Moreover, the proposed system highlights the consideration of balance between the rights and legitimate concerns of elderly residents and the requirements of an efficient functioning in the healthcare facility.

Keywords: fall prevention, healthcare data protection, elderly monitoring, motion sensing, mobile technology.

1 Introduction

In around 35 years and by 2050, it's estimated that more than one in each group of five people will be aged 65 or over. In this age group, falling is one of the most serious life-threatening events that can occur, as approximately one-third to one-half of the population aged 65 and over (mostly aging care centers residents) experience falls on a yearly basis and half of these elderly do fall repeatedly [1]. That is, the increasing aging of the population requires new kinds of social and medical intervention and the availability of different supportive services for elderly people.

Nowadays, medical science is developed in collaboration between human and technology. Medical professionals and caregivers have to follow strict rules for the protection and safeguard of human life. However, this might be subject to secrecy and respect

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 196–203, 2013.

for privacy, nevertheless sensible data is being shared between persons and applications. So, there must be a balance between the rights and legitimate concerns of users and the requirements of an efficient functioning of healthcare facilities [2].

A fall can be defined as unintentionally coming to rest on the ground or other lower level with or without loss of consciousness [3]. Unintentional falls is an ever-increasing problem among the elderly population that presents a common cause of severe injury. However, due to the advances in modern monitoring and detection technologies for old people that report fall events in real time, the average age of global aging population increases continuously in recent years. Aging people are frailer, more unsteady, and have slower reactions, thus are more likely to fall and be injured than younger individuals. Typically, many falls are simply managed using different types of alarm devices to notify others when a falls event occurs. However, it's also essential to offer various practical solutions for improving the quality of life for elderly people along with assisting them and their caregivers against falls. So, falls detection and prevention technologies shall extend beyond simple notification as currently it can be used to screen for falls risk and accordingly to prevent a fall from occurring.

Falling among the elderly happens due to different causes as well as it leads to different consequences. Being aware of those reasons and consequences serves researchers, designers, and developers of fall detection and prevention systems to develop various creative solutions for the problem of elderly falls. Technical staff or caregivers may need to process data and information concerning the users/patients on a daily basis. Due to that fact, they may sacrifice privacy to maintain a flow of data that is quite important for the availability of efficient and trustable healthcare services.

This article brings the concept of elderly healthcare data protection to the light in addition to proposing an automatic video surveillance system for elderly protection via falls detection and prevention. The proposed approach clarifies the utilization of applying the elderly healthcare data protection concept along with the integration of motion detection sensors embedded in network cameras with smart mobile phones in order to develop a platform for pervasive fall detection and prevention. Moreover, the proposed system highlights the consideration of balance between the rights and legitimate concerns of elderly residents and the requirements of an efficient functioning in the healthcare facility. The rest of this article is organized as follows. Section 2 introduces the fundamental guidance rules for healthcare data protection. Section 3 presents the phases of the proposed *FallPrevent* system as well as highlighting system architecture and functionality for elderly fall detection and prevention. Finally, section 4 concludes the article and discusses future challenges.

2 Healthcare Data Protection

Although it is allowable that elderly monitoring and protection systems may respect legal requirements concerning the rights and warranties of the data holder, it must nevertheless be recognized that there are permanent risks of Dataveillance [4]. The dataveillance is the concept of systematic usage of personal data systems in the investigation or monitoring of the actions or communications of one or more persons. On the other side, one must distinguish between requirements concerning privacy and requirements concerning data protection, between warranties of opacity and warranties of transparency [5].

Therefore, it becomes important to understand that it is not enough to have a legal affirmation of rights. It is also quite important to ensure the effectiveness of these rights. Thus, it is quite important to indicate that technology brings along multiple and quite serious threats to the human rights to privacy and data protection as well as playing an essential role for finding the solution for these problems, while enhancing technological uses in compliance with the legal requirements of privacy and data protection [6].

The Data Protection Acts [7], [8] 1988 and 2003, which were defined to guideline the processing of data on identifiable living people, governed the protection of personal data in order to provide data controllers a legal responsibility to:

- obtain and process personal data fairly;
- keep it only for one or more specified and explicit lawful purposes;
- process it only in ways compatible with the purposes for which it was given initially;
- keep personal data safe and secure;
- keep data accurate, complete and up-to-date;
- ensure that it is adequate, relevant and not excessive;
- retain it no longer than is necessary for the specified purpose or purposes; and,
- provide a copy of his/her personal data to any individual, on request.

The purpose of these guidelines is to assist ensuring, as much as possible, that personal data in their possession is kept safe and secure and to help meeting legal responsibilities as set out previously. For information to be processed fairly, the data subject should know who the data controller is, why the information is being processed and any other necessary information, such as the likely consequences of the processing. The purpose for which personal data is collected and processed should be made clear to the data subject. Data subjects should not be deceived or misled as to the purpose for which their information is held or used. Information should only be obtained from a person who is legally authorized to supply it.

For data to be processed legally, it must not lead to any kind of discrimination and should not go against any other laws such as the Human Rights Act 1998. Data controllers and users must not collect and use personal information unless there is a specific and valid reason for doing so. The data subject must be told what the information will be used for. Finally, personal information collected for one reason must not be used for any other unrelated purpose. That is, only information needed for the specific purpose should be asked for or recorded. Information that is not relevant for the purpose must not be collected simply because it might be useful in the future.

3 The Proposed Elderly Protection Approach

3.1 Architecture and Functionality

The proposed automatic video surveillance system for elderly protection via falls detection and prevention; namely*FallPrevent*, employs typical off-the-shelf IP network cameras and *Android OS* smart phones. We developed an *Android OS* application, namely

CameraWatcher that is installed on the smart phones held by the caregivers for providing true around the clock surveillance so experimented residents can be checked on in their rooms at anytime with the Android smart phone mobile device. A WiFi LAN wirelessly connects the wireless cameras and caregivers smart mobile phones to a server sits in the IT support room. Connectivity via WiFi LAN enables obtaining the best visual profile the network camera can give, which is (30 fps) and a camera response time of a few hundred milliseconds [9]. Figure 1 depicts the architecture components of the proposed *FallPrevent* system showing interconnectivity among different system components.

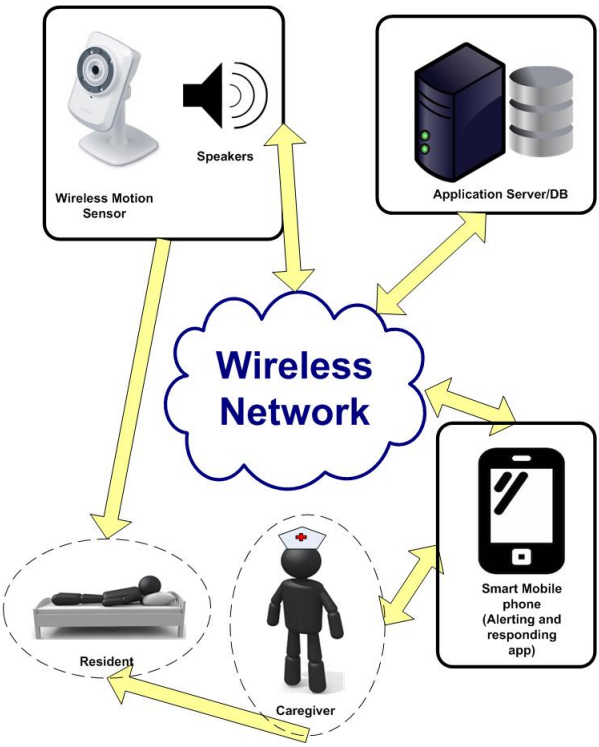


Fig. 1. FallPrevent: System Architecture

The proposed elderly falls prevention system is composed of three phases, as shown in Figure 2. The phases are *Monitoring*, *Movement Detection and Alarming*, *Alarm Confirmation and Fall Prevention*.

During *Monitoring* phase, after installing the network wireless motion sensing units (cameras), above the beds in residents' rooms, vision sensors within those cameras started obtaining motion detection information about the person on the bed. A visual sensor (camera) installed above the bed monitors any movements by the person on the bed in a non-restrictive manner.

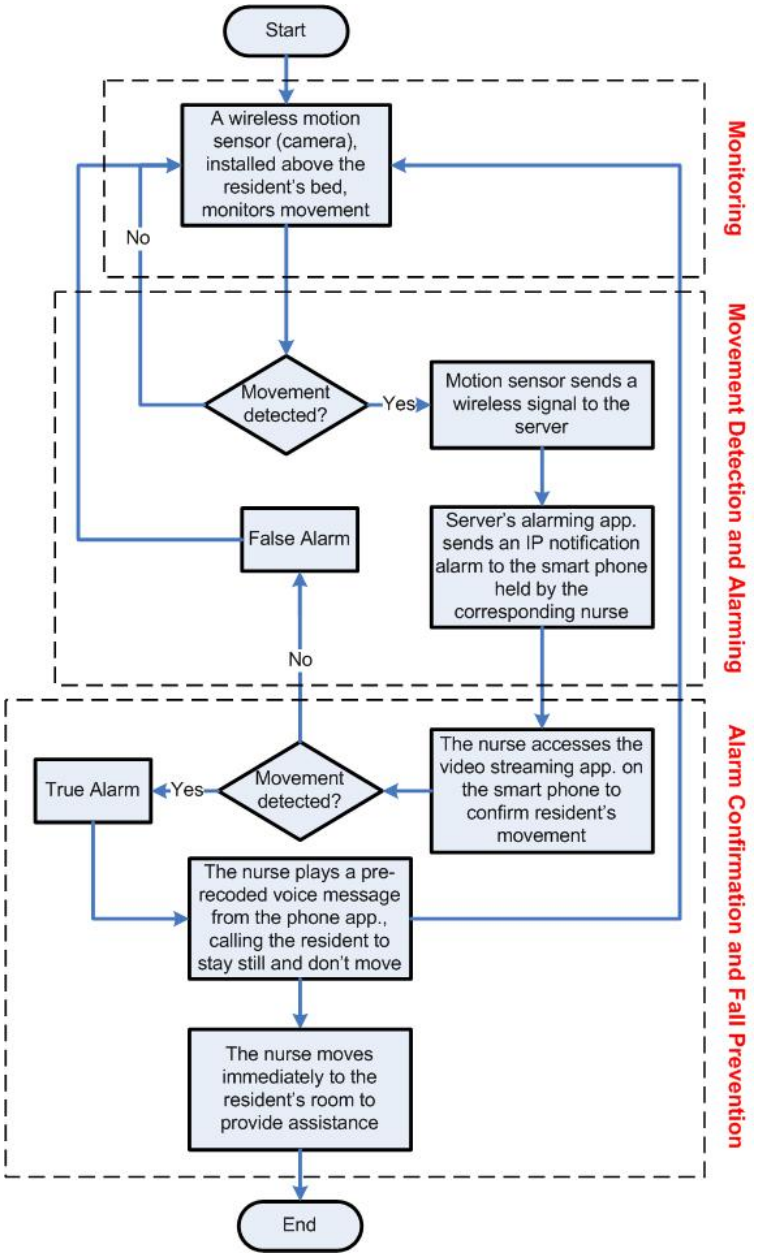


Fig. 2. FallPrevent: System Phases

For *Movement Detection and Alarming phase*, if a fall was suspected (a movement was detected), an automatic IP notification alarm would be sent to the mobile phone of the corresponding caregiver for that room with information including the date, time, Room number, and patient ID or name.

Concerning *Alarm Confirmation and Fall Prevention phase*, on receiving a notification alarm, for avoiding false alarms, the corresponding caregiver accesses the live video streaming option in the smart phone application to confirm resident's movement. For each resident, a voice message has been previously recorded by one of his/her family members in order to be triggered remotely by the caregiver, calling the resident to stay still and don't move, till having assistance.

So, on confirming resident movement (true alarm), the caregiver selects an option in the smart phone application remotely triggering the recoded voice message till reaching the resident's room and providing immediate assistance. Otherwise, the caregiver just cancels the alarm (false alarm) and considers that the resident was not intending to move away from the bed as he/she went back to the still status.

3.2 Experimental Population and Data Protection Application

The targeted population is the elderly residents of the Edmonton ChinaTown Care Center [10] during the first six months of the year 2013 (January to June 2013). A *benchmark dataset* was constructed by focusing on a specific sample of the whole population represented by the elderly residents of 8 rooms existed in the second and third floors (4 rooms at each floor). To protect the privacy of individuals, we excluded and removed all personally identifying information, including the person's name, Social Security Number (SSN), medicare number, and birth date. Data records about monitored residents in the *benchmark dataset* are consisted of [Room_No., Resident_ID, Age, Gender, Medical history, Falls history]. Medical history field contains health-related data about the resident such as being with diabetes, hypertension, heart disease, asthma, cancers, arthritis, visual impairment, mobility disorders, etc. Also, data about medications for each resident is saved in his/her record. Falls history field contains recordings about (number of falls/day, cause, injury type [arm/leg/back], etc.).

Additionally, for protecting residents' privacy, an approval has been obtained, from each one of the elderly residents in the 8 tested rooms. That approval notifies the resident (or his/her family) that a network camera will be deployed in his/her room just above the resident's bed (for experimental reasons) to monitor and detect any movement by the elderly resident at that room and send notification alarm for the corresponding caregiver's mobile phone if any movement detected. The movement sensor embedded into the cameras installed at each one of the tested 8 rooms is pre-programmed to automatically turn on and operate daily (from 7:00PM till 8:00AM) the next day, the case that satisfies privacy protection rule of "only information needed for the specific purpose(s) should be recorded". According to the CEO of the care facility, during the duration (from 7:00PM till 8:00AM), caregivers experience the highest ratio of elderly residents movement tendency and falling accordingly.

Moreover, the *benchmark dataset* contains detailed data records about the history of falling within 6 months officially retrieved from the care center documented records. The *FallPrevent dataset* will be constructed by saving details about alarm events

triggered due to resident's movement on a daily basis. Residents' movement and falls behavior records in the *FallPrevent dataset* will be used for comprehensive assessment for the proposed *FallPrevent* system after operating for 6 months or more.

Hence, it can be concluded that the design of the proposed elderly falls prevention system takes into consideration the healthcare data protection guidelines previously stated in section 2. That is, the participating residents know who the data controller is, why the information is being processed, what the information will be used for, manual records have been legally obtained from the care center employees legally authorized to supply it, the collected data has been processed legally and free of discrimination, no personal data has been collected, collected data will not be used for any other unrelated purpose, and finally only the data needed and previously stated has been recorded and no irrelevant data has been collected as it might be useful in the future.

4 Conclusion and Future Work

Elderly people in long-term care facilities or generally aging persons with cognitive impairment are at high risk of falling and more specialized technology solutions must be developed specifically for these populations. Gathering and sharing data and information between hospitals, healthcare facilities, physicians, caregivers, and other professionals improves the healthcare services. Nevertheless, it cannot be forgotten that privacy and data protection are fundamental rights of the patients.

A number of future directions are currently considered for developing the elderly protection fall prevention system proposed in this article. The suggested developments have to strongly consider the fundamental rights of the patients for privacy and data protection as well. One main challenge for future developments of the proposed elderly fall prevention system is via enabling RFID based tracking for caregivers with considering their status (busy/idle). That is, to incorporate an additional RFID based localization and tracking module with the proposed system. That approach will enable the system to assign and alarm the idle caregiver whose location is the closest to the room of the resident in risk, without assigning certain caregiver to specific room(s).

Acknowledgment. Thanks to the Mitacs-Accelerate Internship Program and the industry partner, Remote Transportation Solutions Ltd.

References

1. Tinetti, M.E., Speechley, M.: Prevention of Falls Among the Elderly. The New England Journal of Medicine 320(16), 1055–1059 (1989)
2. Ângelo, C., Andrade, F.C.P., Novais, P., Simoes, R.: Privacy and data protection in elderly healthcare: threats and legal warranties. In: Proceedings of Sixth International Workshop on Jurisinformatics (JURISIN 2012), Japan (2012)
3. James, K., Eldemire-Shearer, D., Gouldbourne, J., Morris, C.: Falls and Fall Prevention in the Elderly: The Jamaican Perspective. West Indian Medical Journal 56(6), 534–539 (2007)
4. Clarke, R.: Information technology and dataveillance. Communications of the ACM 31, 498–512 (1988)

5. Hert, P., Gutwirth, S., Moscibroda, A., Wright, D., González Fuster, G.: Legal safe-guards for privacy and data protection in ambient intelligence. Personal and Ubiquitous Computing 13, 435–444 (2008)
6. Winn, J.K.: Technical Standards as Data Protection Regulations. In: Gutwirth, S., Poullet, Y., Hert, P., Terwangne, C., Nouwt, S. (eds.) Reinventing Data Protection?, pp. 191–206. Springer, Netherlands (2009)
7. The Data Protection Commissioner: Protecting the confidentiality of Personal Data: Guidance Note. CMOD, Department of Finance (December 2008)
8. Guide to Data Protection - Data Protection Act 1998 - ICO. Information Commissioner's Office (1998)
9. Issa, O., Grégoire, J.-C., Belala, Y., Wong, J., Bage, M.: 3G Video Uploading Applications: Performance and Improvements. IEEE MultiMedia 15(4), 58–67 (2008)
10. Edmonton Chinatown Care Centre (2006), `http://www.edmccc.net/` (accessed September 2012)

A Secure Framework for OTA Smart Device Ecosystems Using ECC Encryption and Biometrics

Miguel Salas

Intel Corporation, Microprocessor Group, Fort Collins, Colorado, USA
miguel.o.salas@intel.com

Abstract. As we move towards a world where all the traditional household appliances and basic industrial devices are being transformed into interactive high-computing devices, an ecosystem of these smart devices is emerging. With this impending revolution, often coined the *Internet of Things*, one of the understated challenges is the security infrastructure that must accompany the deployment of this ecosystem. In this paper we propose a security framework that leverages hierarchical hardware memory mapping, modularity of the Operating System, and an efficient biometric aided ECC cryptosystem to work together towards this security need. We focus on the secure and efficient implementation of OTA updates and inter-device communication. Our work shows that by integrating several novel improvements based on real system considerations with state-of-the art techniques, we can build a commercially feasible security framework for these devices that is 35% faster and 5% more load efficient than current state-of-the-art ECC-based cryptosystems and OTA compression schemes.

Keywords: OTA, security framework, biometrics, elliptic curve cryptography, Internet of Things.

1 Introduction

While the use of smartphones is already ubiquitous and the number of software-controlled electronic components in a car continues to increase at a fast rate, we have yet to see a widespread deployment of other devices capable of smart computation and high user interaction. These devices range from printers and TVs, whose smart versions are slowly being introduced to the market, to consumer electronics, such as refrigerators and thermostats whose smart versions are yet to achieve a significant consumer base.

While these smart devices lack strong connectivity among each other, their design is trending towards high performance computing with innovative capabilities and high interconnectivity among them. For example, envision a smart shower system, a thermostat with access to the outside weather, and a smart car, all Wi-Fi enabled. If the smart shower system learns that the user will want to drive his car within 10 minutes of getting out of the shower, it can query the thermostat to determine whether the weather outside is below a certain temperature, it will request the car engine to start as soon as the user is out of the shower. Similarly, a smart fridge can keep track of all

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 204–218, 2013.

items stored inside, and if a grocery item is running low, it can query the closest laptop in the house to place an order on an online grocery-shopping site. In this manner, these devices can create a *smart device ecosystem*. Figure 1 illustrates an ecosystem of this nature.

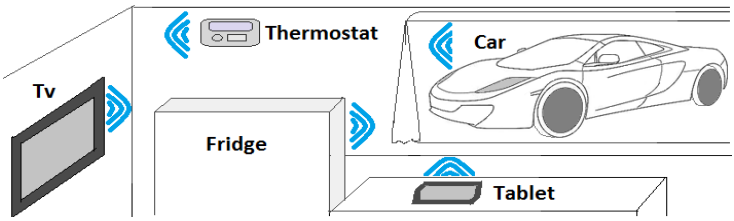


Fig. 1. Sample ecosystem of smart devices of the future

Connectivity enablement for these ecosystems has already been recognized as a need [1], and has been the leading motivation to create Operating Systems such as Tizen [2], which can enable a web API based connectivity infrastructure among all these devices. However, while providing a natural next step towards smart device evolution, these ecosystems make attractive targets for security breaches. Going back to the examples above, a malicious attacker can now masquerade as the shower system, creating a replay attack to turn on the car engine whenever he desires, or masquerade as the fridge to charge any grocery items it pleases to the user's credit card. Hence, security for such ecosystems is imperative.

In this work, we propose a framework to address this security need. Unlike the previously proposed security frameworks, our work leverages three key unique aspects of these devices: (1) Unlike nodes on a WSN, these devices are expected to have some level of user interaction; (2) Unlike mobile phones, because these devices live within a geographically-limited network, we must consider efficient intra-net communications among them; (3) Given the known software and hardware deployment patterns of these devices, there are optimization areas at the software-hardware integration level. Our work proposes a strong, efficient, and flexible security framework, which takes into account unique aspects of smart devices including their physical memory layout, the patterns of OTA update deployments, and the need for ergonomic and cost-efficient designs.

The rest of the paper is divided into seven sections. Section 2 discusses previous work on the different areas that make up security infrastructures of this nature and Section 3 discusses the high level view of the proposed infrastructure. Section 4 contains the description of the framework, including the hierarchical OS design, OTA transmission protocol, proposed encryption algorithms and the biometric sensor integration. Section 5 presents the system performance results and Section 6 concludes the paper.

2 Related Work

The imminent arrival of the Internet of Things (IoT) infrastructure is best envisioned in [1]. Any IoT device must possess an OTA update capability. Several OTA-enabled frameworks have been proposed. The essential OTA enablement steps are outlined in

[3]. The OTA-PSD framework [4] shows the underlying mechanism from the user and provider necessary for OTA updates, including the XML based software versioning scheme. The OTA-WNS framework [5] shows how to model a device's state as an OTA-update-centric state machine.

Furthermore, any OTA framework needs security, as outlined in [6]. The security extension for OTA-enabled devices proposed in [7] relies on the use of the TLS extension over TCP packets. Similarly, the SenseOP framework [8] makes use of ECC encryption to provide security guarantees assuming tamper proof device interfaces, which do not always hold. In [9], a hash chains based security framework for OTA updates on smart cars is proposed, although it is computationally light, it relies exclusively on tamper proof devices that make use of private keys. Similar to the SenseOP protocol, such assumptions will create single point vulnerabilities in IoT infrastructures. Our framework does not rely on these assumptions.

An OTA-enabled IoT device also needs to take into account the process of selecting the data for the OTA update. Since an OTA update must use network bandwidth, we are concerned with minimizing the transmission data and latency required. Recently proposed efficient solutions such as The Two-step Differential [10] and Queen Differential [11] are schemes that transmit only the difference between the current and new software and firmware version. Regardless whether such differential schemes are applied, typically the data is compressed. Conventional compression schemes are not ideal in this model, since the computing should be loaded onto the server whenever possible. In contrast, Byte Pair Encoding, explored in [12] and [13], is an asymmetric technique fit for this scenario.

Our framework relies on an improved version of Queen Differential, QDiff. The raw QDiff scheme relies in the two-stage differential process shown in figure 2 [10]. While QDiff can achieve small deltas in the image, it does so with some drawbacks. First, its space pre-allocation creates some slop spaces when existing code is deleted. This has the side effect of fragmentation over the long term. Furthermore, once it has found the delta it sends the delta over the network without specifying a compression scheme. Moreover since QDiff algorithm runs in exponential time with the size of the code in question, a large code database can make its runtime prohibitively large. Finally, QDiff is not optimized towards exploiting the lifecycle patterns of the OTA updates, which will be discussed in section V.

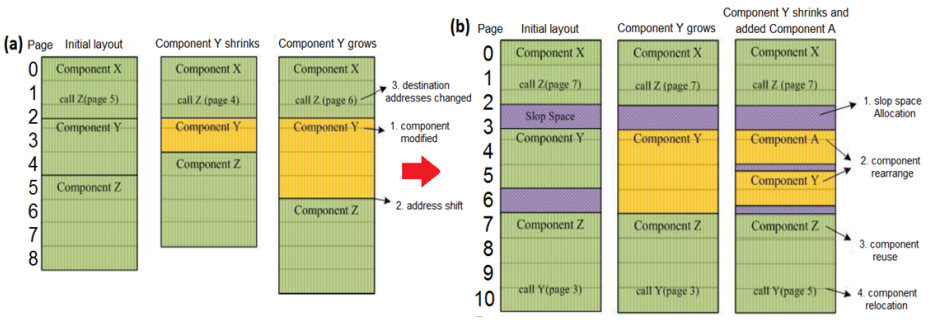


Fig. 2. Sample addition/deletion of new code for small-delta OTA updates using two-stage differential schemes such as QDiff. (a) Shows an update when a memory component (Y) grows. (b) How the scheme attempts to minimize component delta and slop space.

Finally, our framework relies on public cryptography because the devices are inherently heterogeneous and hence do not share a secure channel. Among public cryptosystems, the RSA public cryptosystem is the most widespread protocol, relying on the difficulty of factoring large numbers and finding discrete logarithms. According to the NISA, we can achieve a similar security offered by RSA cryptographic systems with Elliptic Curve Cryptography (ECC) through the use of a much smaller key size. A comparison of key sizes is shown in table 1 [14]. The performance advantages of ECC over RSA are outlined in [15], and security frameworks for mobile devices based on ECC are proposed in [16] and [17]. Note that as specified by the NISA, this relies on choosing a strong ECC curve.

Table 1. Equivalent security strength guarantees between symmetric key algorithms, RSA and ECC, by key sizes

Symmetric Key Algorithms	RSA	ECC
80	$k = 1024$	$f = 160\text{-}223$
112	$k = 2048$	$f = 224\text{-}255$
128	$k = 3072$	$f = 256\text{-}383$
192	$k = 7680$	$f = 384\text{-}511$
256	$k = 15360$	$f = 512+$

Finally, the key to our scheme is the use of biometrics. However, the use of biometrics to authenticate users typically suffers from accuracy issues. Proposed solutions can be divided into multi-sensor devices [18][19][20], whereby two or more biometric sensors are taken as inputs to improve accuracy, or multi-algorithm, whereby two or more algorithms are run independently based on the same biometric sensor. The former approach suffers from poor ergonomics: asking a user to first log in with a face read, then with a fingerprint, and finally with a retina reading might yield a high-accuracy, high-security system, but it would be far from practical for consumer devices. The latter approach suffers from the limitations of the biometric sensor itself. If a biometric sensor reads a fingerprint and takes several patterns of it, the finger itself can become a single point of failure. Instead, our approach is a simple and ergonomic solution based on a hybrid multi-sensor approach.

3 High Level Overview

In the smart device ecosystem, there are three forms of communications: (1) from the device to the OTA provider, (2) among the smart devices inside the ecosystem in an ad-hoc wireless network, and (3) from the devices to other HTTP servers on the internet. This is shown in figure 3(a). In this paper we are exclusively concerned with (1) and (2) because it is assumed that (3) can be handled using HTTPS requests and depends uniquely on the HTTP server policies of which we have no control over. Thus in our framework we optimize towards those two communication paths. Furthermore we model communication between the manufacturer and the device to be exclusively OTA updates. We assume that the hardware manufacturer and the operating system provider act as a joint entity, henceforth referred to as the manufacturer, for device updates. Therefore, we assume that OTA updates concern all of the ROM contents: the operating system, the firmware, and immutable native user applications. Therefore, in this paper, any OTA update reference is interchangeable with OTA-ROM updates. The high-level scheme is depicted in figure 3(b).

Our contributions are as follows. We exploit hierarchical memory mapping for OTA benefits and encrypting a secure key. Furthermore, we propose leveraging the software update versioning scheme in use today for OTA updates. Finally, we propose a novel fingerprint-based biometrics to provide user authentication and aid the ECC encryption protocol.

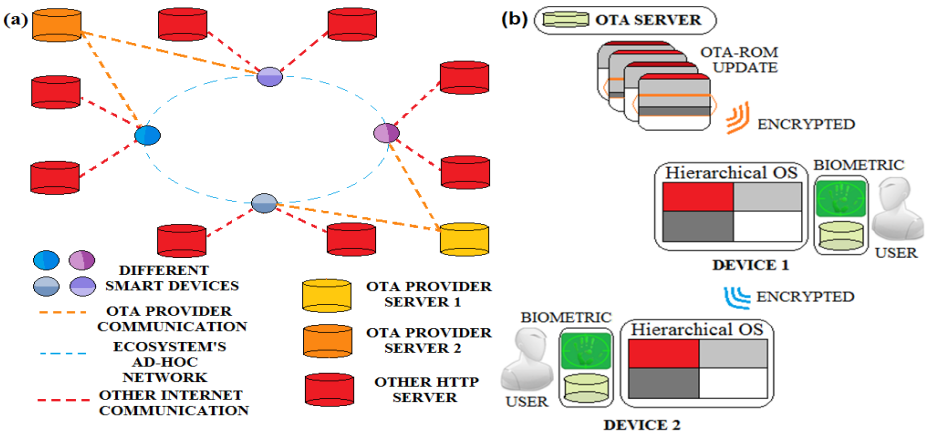


Fig. 3. (a). High level overview of the framework. The three types of communication in the smart device ecosystem: (1) from the device to its OTA server, (2) intra communication, and (3) from the device to other HTTP servers. Notice that we assume heterogeneous devices and providers. **(b)** High level model of the proposed security framework focusing on (1) and (2)

4 Security Framework Description

Our framework consists of four different components: a hierarchical memory mapping for OTA updatable components, an OTA protocol, an encryption scheme, and a biometric sensor support. Each of these is described in detail below.

4.1 Operating System Support

We propose the use of a modular Operating System and its firmware components so that hierarchical memory mapping in the ROM is achievable. We propose dividing up

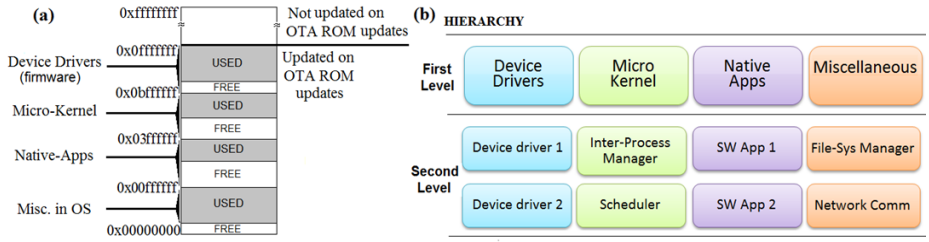


Fig. 4. Memory organization for the OTA components **(a)** Shows the proposed memory partition with sample memory addresses. **(b)** Shows a two level hierarchy that although one could use to partition the memory would result in severe fragmentation.

the ROM into a four component hierarchy, so that ROM updates can be treated separately for each of these four hierarchies. Figure 4(a) shows the proposed division.

Note that we arbitrarily choose three major hierarchies and allocate a fourth hierarchy for the rest. Although we could extend this to a two-level hierarchy as explained conceptually in figure 4(b), we do not do so because more partitions are likely to cause code fragmentation and reduce the ability of the compiler to allocate memory space efficiently.

4.2 OTA Update Protocol

The trend for smart device development, from game consoles to smartphones, is to release software and firmware OTA updates backwards compatible with the devices up to a target device life cycle. For example, Android and iOS devices can continue receiving firmware updates from the device manufacturer up to a few years out in time, after which the device is no longer supported. The target device life varies sharply among devices, with gaming consoles lasting roughly a decade. However, software and/or firmware OTA updates follow a cyclical pattern such that there is a major upgrade, encompassing new device features, followed by several minor firmware updates, typically targeted towards bug fixing. Figure 5 illustrates the typical firmware OTA update deployment cycle. As mentioned in section 2, in our framework, we propose using a modified version of the Queen Differential (QDiff): the *Modular and Cycle Aware extended Queen Diff* (MCA-QDiff).

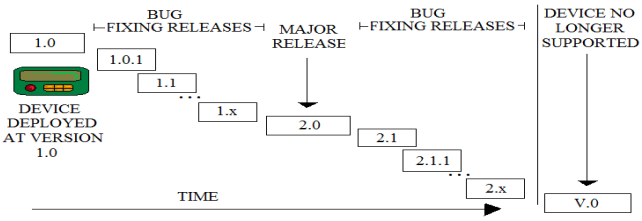


Fig. 5. Typical firmware update lifecycle for smart devices, including minor and major releases

The modularity of the OS memory mapping enables MCA-QDiff to operate at a finer grain than the original QDiff by enacting the code delta on a ROM hierarchy only if it was marked as touched. To obtain this functionality, at the time that an OTA is deployed, a header packet contains information about whether a ROM hierarchy will receive an update or not. Furthermore, because the order of the ROM is hierarchical as described in section 4.1, the area of the ROM that each of these hierarchies occupy is bounded by address ranges. This allows the code delta to be applied only at that level of hierarchy. For our proposed infrastructure, there are four hierarchies living in the OTA updatable ROM image: micro-kernel, native software applications, device drivers, and miscellaneous functions that do not fall into any of the previous three categories. Inside any of these hierarchies, memory boundaries blur and we let the compiler produce an efficient image implementation. Once a hierarchy in the ROM has been marked as needing update by the header packet, it applies the raw QDiff approach.

As mentioned in section 2, QDiff adds jump instructions to link the current code with the new code and pads where the old code is deleted, thus minimizing compile image differences. However, conducting this procedure repeatedly can create excessive padding in the gaps of the erased code leading to code fragmentation, and is especially inefficient when the code deltas are very large. In fact, there is nothing stopping the raw QDiff to produce an image delta that is larger than the whole image itself. However, because we now have an idea of how the ROM updates are being performed, we know that it is not reasonable to send a compressed image when it has had major code overhaul. So instead, MCA-QDiff chooses to send the complete image of a given hierarchy when such overhauls happen, and compute the ROM code delta only on minor updates. Note that even on major updates, the full image is not sent. For example, if the Micro-kernel and device drivers stayed the same, yet all else changed, it would send a header stating so, along with the changes for the other two chosen hierarchies.

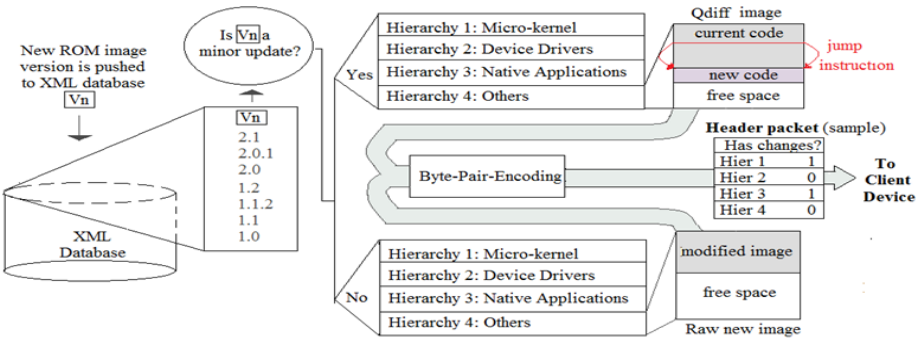


Fig. 6. Proposed MCA-QDiff transmission protocol for OTA ROM image updates.

Whether it was a minor update with a minimal image delta produced or a major update with the selectively sent full image, it is compressed prior to sending using Byte Pair Encoding as discussed in section 2. This yields a small image for the minor firmware version releases, which is asymmetrical in computing time: it takes much longer to encode at the server than to decode inside the device, meeting our design goals. Finally, we leverage the XML based approach to keep track of the ROM image versioning [4]. The entire OTA transmission protocol is shown in figure 6.

4.3 Encryption Protocol

Due to the high number of opportunities for malicious attacks, encrypted communications are a must in this ecosystem. Previously proposed approaches have made use of the fact that one can embed a secret key on the hardware ROM itself, often leveraging the serial number in the device and storing it in the manufacturer's server database. This approach has two major drawbacks: (1) all encrypted device communications must be between the device and the manufacturer's server that possesses this key, and (2) anyone able to compromise the server, can now masquerade as the server to access information on the target device. Thus, we turn to public key encryption to bypass

these challenges at the cost of increased overhead in the communications. We argue, however, that since this overhead is only paid when establishing new communication sessions, it is tolerable, given its higher flexibility, resilience, and security. Furthermore, as explained in section 2, we choose public cryptography based on ECC over RSA due to its smaller key size. ECC is also a computational asymmetric protocol: calculating a computing intensive elliptic curve can be done on the server side, while the rest can be done on the client side. Hence, we rely on ECC for encryption.

To improve performance, our framework attempts to leverage the secret key channel established by the manufacturing stage process. In our framework, the manufacturer will install a set of private keys in the device's ROM memory. We choose a 224 bit ECC system because it is equivalent to the 2048 bit key RSA. Notice that a 224 bit key requires only 7 32-bit registers in RAM for operation, while a 2048 bit key RSA requires 64 32-bit registers, which is roughly the size of an entire register set found in simple low-power microprocessor architectures. Thus, calculations involving RSA keys would slow down hardware performance significantly.

Private Key Protection. A public key cryptosystem with a compromised private key is no longer reliable. For this reason, our framework takes into account the two main security risks in private key deployment: (1) attempting to compromise the ROM location where the keys are stored with a bit flip, and (2) attempting to extract the key from the ROM location without leaving a trace of device tampering. Note that the former also covers reliability, since alpha particles or manufacturing defects can lead to bit flips.

To address the bit flip concerns, we create a redundant copy of the key in the ROM at manufacturing time. Because we are trying to address bit flips, we cannot store the key copies in adjacent memory locations, due to the locality principle of manufacturing defects and typical ROM attacks. However, we cannot place them at random memory locations as this would create unnecessary code fragmentation. Therefore we leverage the memory hierarchy proposed in section 4.1 by placing the redundant keys at the start of each of these hierarchies. To address device tampering on the secret key, the manufacturer will not just install a key, but instead create two 112 bit keys which will be concatenated to create the 224 bit key. Because we assume keys are stored in pages with 32-bit addressing, the device simply needs 4 memory locations for each of the two 122 bit keys, with the last memory location per key padded with 16 bits of random data. Such system avoids an unnecessary XOR or more complex hash operation to produce a key based on two other keys at run-time. We assume that while a malicious user might snatch one of the 122 bits private keys, it is unfeasible that he might find both and know to concatenate them.

Finally, since we need a dual key system for the reliability concern and a dual key system for the tampering concern, we use a set of four 122-bit keys, which match seamlessly with the four hierarchies used in the memory mapping. The full private key installing scheme, which we refer to as the Memory Hierarchy based Private Key (MHPK) scheme is illustrated in figure 7. Note that our system can detect bit flips and key tampering because after the first set of keys do not yield successful authentication, it will attempt its second set of keys. In the extremely rare case that the second key also fails, to avoid leaving the device in an obsolete state, we rely on biometric user authentication to request a ROM update from the server, as will be discussed in section 4.4.

Session Key Generation. While having a reliable private key in the device is the first step towards an ECC cryptosystem, it is far from enough. In our framework, the manufacturer will install the device with its public key as well. This requires calculating the Elliptic Curve math at install time. The server will calculate a strong elliptic curve, pick a point on the curve and come up with the device's public key based on the curve parameters. Once the device public key has been determined, the manufacturer will install the device's public key and the manufacturer's public key in the ROM of the device. This public key pair will not have the same protection scheme as the private key. When the device is deployed, it will now be able to calculate the session key from the start up, based on its own private key and the server's installed public key. Finally, note that it also installs the elliptic curve parameter p, a, b into the device so that the device will be able to just pick random points on the curve to generate different public keys later on, but will not waste resources computing another strong elliptic curve.

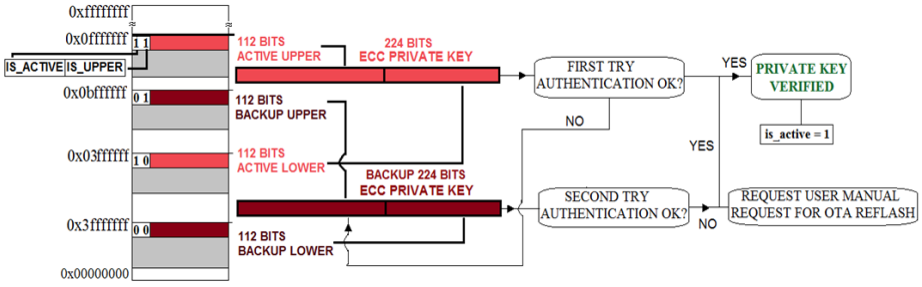


Fig. 7. Memory Hierarchy based Private Key MHPK scheme. In this scheme, we use four 112 bits to protect against malicious attacks in producing a trusted 224 bit key for ECC encryption. After using up the spare key, the device will request user input to re-flash the contents.

Notice that the server only keeps the record of the device's public key but it discards the device's private key, yielding an attack on the server useless against the device itself. Finally, a packet sequence number, which resets after 2^{32} bits for ease of implementation, is used as cryptographic nonce to protect against replay attacks.

Authentication. Authentication protects the devices from man-in-the-middle attacks, as well as providing integrity checks. While typical authentication protocols with public key cryptosystems would require the use of a trusted third party to verify that each other's public key is indeed what is claimed, we choose to not use certificates due to the large overhead of public key infrastructure such as the use of third party certificates. As discussed in section 3, we only concern ourselves with addressing communication between a device and its manufacturer server, and between two devices inside the ecosystem. We choose to authenticate these connections by using a hash-based message authentication code. The HMAC's secret key input is the user's biometrics, whose mechanism will be discussed in detail in the next section. A strong but hardware efficient HMAC like the 128-bit SHA-3 [22] is suggested. Note that a system that authenticates devices belonging to the ecosystem using a secret key generated by the user biometrics can now block man-in-the-middle attacks because a malicious user without access to this key cannot masquerade as one of the devices

belonging to the ecosystem. Likewise, we use this key to digitally sign messages for inter-device communications and between the devices and their OTA provider. Note this supports heterogeneous devices coming from different OTA providers as it is likely to be the case for commercially deployed smart device ecosystems.

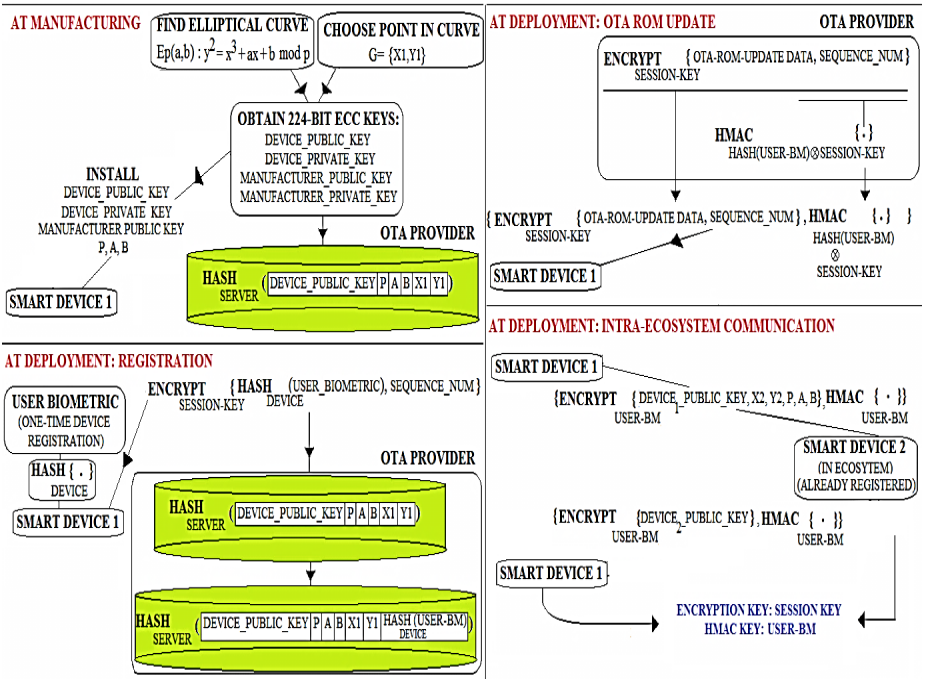


Fig. 8. Device cryptosystem. Notice that it embeds the elliptic curve at installation time. It only encrypts with the ECC session key and provides authentication and integrity guarantees using a HMAC whose key is the hash of the user biometric, common to all ecosystem devices. The HMAC key is the user biometric, common among all the user's devices.

Table 2. Denial of Service protection provided by the framework. Note that when there are only a few failed attempts, the wait time penalty is tolerable, taking into account legitimate failed attempts, but is unforgiving to large numbers of failed attempts.

CONSECUTIVE FAILED MASQUERADING ATTEMPT	WAIT TIME UNTIL NEXT TRY (seconds)
1	2
2	4
3	9
4	16
10	1024 (17 minutes)
100	$1.26e^{10}$ (>>1 year)

Furthermore, note we chose to use an ECC encryption scheme, instead of a symmetric key encryption protocol based on the device private key because the devices will have the different private keys with different OTA providers, and can only communicate with their own OTA server. Similarly, we did not rely on a symmetric key based on the user biometrics because if the biometric reading device were tampered with, it would mean that every device in the ecosystem is hopelessly compromised.

The entire protocol is depicted in figure 8. Finally, to protect against denial of service attacks, unlike the approach in [8] with a linear wait time DoS protection, our algorithm will double the response time after every failed authentication attempt as depicted in table 2, which contains a binary factor for efficient hardware implementation.

4.4 Biometric Authentication

While ASCII-password-based authentication schemes remain the most popular way to let users take control of their systems, economical and practical biometric based systems are starting to be deployed. Biometric based systems are attractive from a user convenience perspective, but they pose unique challenges: (1) If a user's raw biometric data is compromised, there could be cultural and legal repercussions, (2) biometric devices tend to not be 100% accurate, with the possibility of false positives and false negatives, and (3) the user's biometric data must be stored in the device in a protected manner to avoid the compromise of this data.

Instead of the proposed approaches discussed in section 2, ours is a hybrid multi-sensor multi-reading approach that is both ergonomic and commercially viable. Instead of using heterogeneous biometric sensors, we only use fingerprint readings, one of the most economical biometric sensors. In our scheme, the user places all of his five fingers on a palm-sized finger-reading platform, and the system takes at least 3 successful fingerprint readings from each of the fingers. This is because it requires the fingerprint data to be consistent before accepting it, as the rest of the devices will rely on this. Then, after having chosen the fingerprint data, it applies a one-way hash creating a key which will be used for the authentication scheme described in section 4.3.

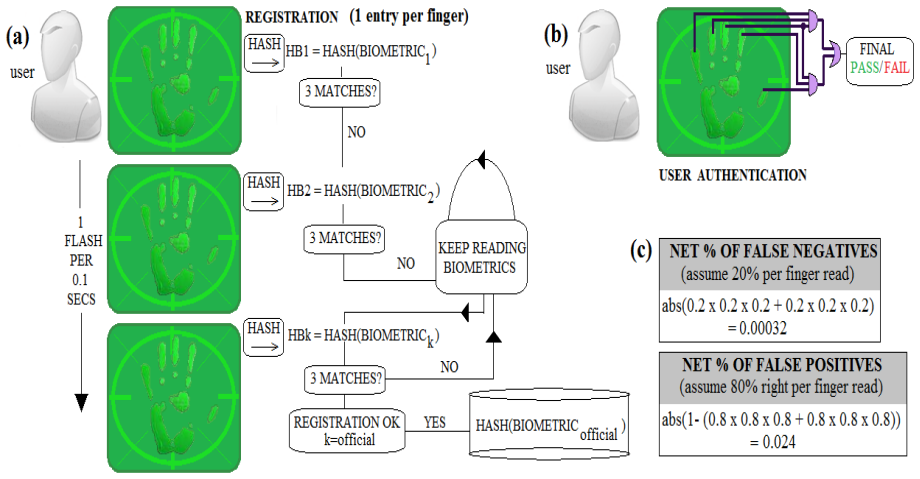


Fig. 9. Registration and user-authentication schemes using economical and ergonomic biometrics. **(a)** For the user registration step, it stores the hash of the user biometric after a reliable result has been found. **(b)** For the authentication step, it performs the AND-OR function. Note its scheme can bump a 20% false negative and positives into a system with negligible failure rates, as seen in part **(c).**

Finally, it allows a user to control the device settings and manual OTA ROM updates by authenticating user manual activity with a scheme that can yield false positives and negatives with near zero probability. The results from first, second and third individual fingerprint reads are AND-ed, while in parallel, the results from the third, fourth and fifth fingers are also AND-ed. Then, these two results are OR-ed together to get the final verdict. Our full biometric based scheme is shown in figure 9. Note that to protect against a malicious user trying to obtain the user biometric data, we never store the data in the device in a raw fashion. Instead, we perform a one-way hash for key generation, and store this same hash to be used for user authentication as well. To verify user authentication, we take the input data, perform the same hash and compare hashes inside the device. Such scheme protects against user privacy attacks on the raw fingerprints themselves.

5 Results

We tested our protocol using a high level simulator that sets up a client acting as the device and two servers, one simulating the manufacturer's OTA provider and the other acting as another device in the ecosystem. Finally, we set up another server acting as a public key infrastructure provider for certain comparison schemes, as will be discussed below. We simulate on four machines, all using Linux centOS 6.4, an Intel Xeon quad core running at a nominal frequency of 2.8 GHz, and 8 GB of RAM. To measure a consistent amount of data sent over the network, we send images made of solely ASCII characters to guarantee a deterministic 1 byte per character ASCII encoding at the network interface. To create more accurate performance comparisons, we maintain the same type of data structures in the code, including encryption and compression algorithms. We measure the performance of our scheme first by measuring the OTA encoding efficiency in terms of the net number of bytes sent and latency for each scheme over a sample software update cycle made of six software updates. We then measure our proposed cryptosystem against an unencrypted system, the de-facto RSA based cryptosystem with a public key infrastructure (PKI), and a similar ECC based cryptosystem with a PKI, using CPU load and latency as the metrics.

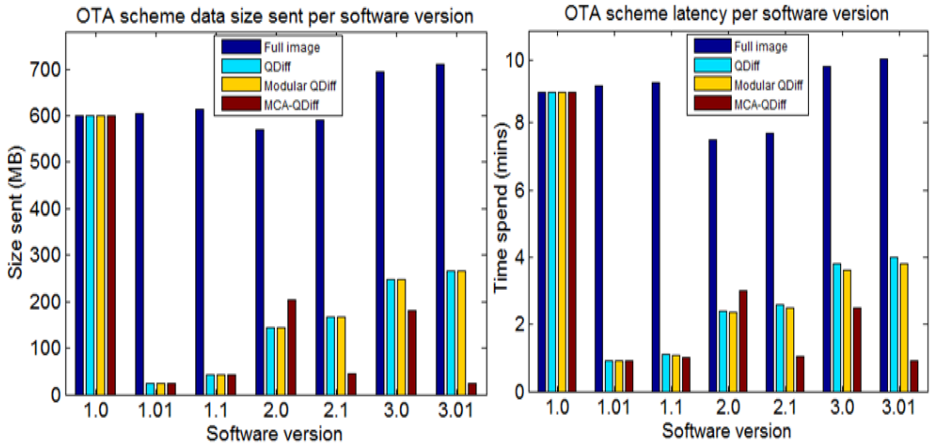


Fig. 10. Comparison of OTA compression schemes: no compression, QDiff, Modular only QDiff, and MCA-QDiff

We first simulate the performance of our OTA protocol against the behavior of the raw QDiff using 600MB images as an input, roughly the expected size of a full OS update [23]. For consistency, all schemes are sent using a biometric-aided ECC encrypted scheme, and all QDiff-based schemes contain a BPE compression step. Figure 10 contains the performance of the OTA protocols given the versioning scheme pattern, comparing four possible scenarios: (1) sending the full image per release, (2) using QDiff, (3) using the *modular* memory mapping enabled QDiff algorithm with a reduced search space, and (4) using our full MCA-QDiff scheme. Note that the sent bytes do not scale perfectly with transmission delay per byte because QDiff based schemes spend time compressing and decompressing with BPE. Furthermore, notice that modular QDiff performs just as QDiff for small deltas, but for larger code deltas, Modular QDiff outperforms the raw QDiff. Finally, as new software versions come out, the code deltas found by QDiff steadily increase, due to lack of self-cleaning and accumulated slop spaces. It is only after the device has gone through enough changes that the advantages of MCA-QDiff become significant. MCA-QDiff obtains an average latency performance improvement of 15% over a sample update cycle.

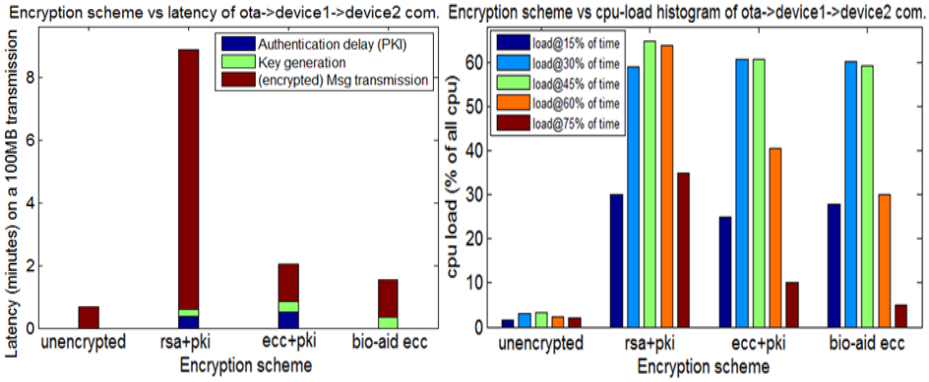


Fig. 11. Comparison of cryptosystem schemes: Unencrypted system, a standard 2048 bit RSA-encrypted system with a PKI, a 224 bit ECC-encrypted system with a PKI, and a 224 bit ECC encrypted system using the biometric for signature instead of a PKI.

To compare the efficiency of our cryptosystem, we implemented four different schemes: (1) sending the unencrypted image; (2) sending the encrypted image using the RSA protocol with a PKI; (3) sending the encrypted image using ECC with a PKI; (4) and finally, sending the encrypted image with our ECC method that bypasses the need of a PKI. We choose to send a 100MB image, the typical size of an OTA iOS update [23]. To emulate a public key infrastructure, we require that each communicating party contact the fourth sever, establish an independent encrypted channel, and have that server search a very large database for a matching key. Notice that when implementing an algorithm, we do so efficiently by exploiting several modulus operation rules, Euclid's algorithm and Fermat's test for compositeness [24], among others. Furthermore, when choosing an ECC curve in our implementation, we pick an arbitrary elliptic curve. However, note that the NST has published a set of "safe" curves [25], and hence on a real implementation, they would be preferred. Figure 11 shows

the full comparison in terms of absolute latency and through a CPU utilization histogram captured at five points during the execution of the simulation. Note that while the unencrypted image has the lowest load and latency on the system, its lack of security is intolerable. Our scheme shows the best balance between performance and security. In particular, our scheme shows a 25% latency reduction over a vanilla ECC system, and a 5% lower average load. Coupled with a 15% latency reduction in our OTA protocol, we obtain a net 35% reduction in latency. Finally notice that our biometric-aided scheme is not only faster, but will also have several other benefits not accounted for in this test bench because a typical public key infrastructure requires additional hardware use and protection schemes, since it is an added source of vulnerability.

6 Conclusion

This paper presented an integrated security framework for a smart device ecosystem. This ecosystem poses unique challenges because the devices are interconnected in a similar manner to wireless sensor networks yet their OTA-ROM provider model is similar to that of smart phones. Furthermore, they have unique characteristics such a level of user interaction much less prominent than smart phones, and yet much more so than conventional WSNs. Updating their software and firmware is done over the air, adding another layer of vulnerability but also opening several optimization areas. Thus, we have presented a system to address this imminent security challenge through design modularity, hardware and software co-design, and finally, through computation and cost efficient design choices. As these smart devices become widely deployed, they will start forming ad-hoc networks on household and industrial settings, meanwhile potential security breach points as well as their treat level will increase exponentially. Hence, designers looking to successfully deploy these ecosystems should turn towards a strong, yet efficient, security framework.

References

1. Vermesan, O., Friess, P., Guillemin, P.: The Internet of Things - Strategic Research Roadmap. In: Cluster of European Research Projects on the Internet of Things, CERP-IoT (2009)
2. Linux Foundation: Tizen OS (2012), `https://www.tizen.org/`
3. Oommen, P.: A Framework for Integrated Management of Mobile-Stations Over-the-Air. In: IEEE/IFIP International Symposium on Integrated Network Management Proceedings (2001)
4. Cong Vo, C.: A Framework for Over the Air Provider-initiated Software Deployment on Mobile Devices. In: 19th Australian Conference on Software Engineering, ASWEC (2008)
5. Ling, Y., Tiansheng, H., Caixing, L., Yue, X., Haoen, Z.: A reprogramming protocol based on state machine for wireless sensor network. In: International Conference on Electrical and Control Engineering, ICECE (2010)
6. Brown, S., Sreenan, C.J.: A New Model for Updating Software in Wireless Sensor Networks. IEEE Network, 42–47 (2006)
7. Bing, B.: A Fast and Secure Framework for Over-the-Air Wireless Software Download Using Reconfigurable Mobile Devices. IEEE Communications Magazine, 58–63 (2006)

8. Bauer, J., Bieling, J., Bothe, A., Schwamborn, M.: Selective and Secure Over-The-Air Programming for Wireless Sensor Networks. In: 21st International Conference on Computer Communications and Networks, ICCCN (2012)
9. Nilsson, D., Larson, U.E.: Secure Firmware Updates over the Air in Intelligent Vehicles. In: IEEE International Conference on Communications Workshops, ICC Workshops (2008)
10. Chiang, M., Lu, T.: Two-Stage Diff: An Efficient Dynamic Software Update Mechanism for Wireless Sensor Networks. In: IFIP 9th International Conference on Embedded and Ubiquitous Computing, EUC (2011)
11. Bin Shafi, N., Ali, K., Hassanein, H.S.: No-reboot and Zero-Flash Over-the-air Programming for Wireless Sensor Networks. In: 9th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks, SECON (2012)
12. Shibata, Y., Kida, T., Fukamachi, S.: Byte Pair Encoding: a text compression scheme that accelerates pattern matching. Technical report DOI-TR-161, Kyshu University (1999)
13. Kiyohara, R.: A New Method of Fast Compression of Program Code for OTA Updates in Consumer Devices. IEEE Transactions on Consumer Electronics, 812–817 (2009)
14. Barker, E., Barker, W., Burr, W.: Recommendation for Key Management. Part 1: General, NIST Special Publication 800-57 (2007)
15. Gupta, K., Silakari, S.: ECC over RSA for Asymmetric Encryption: A Review. IJCSI International Journal of Computer Science Issues, 370–375 (2011)
16. Ganesan, S.: An Efficient Protocol for Resource Constrained Platforms Using ECC. International Journal on Computer Science and Engineering, 89–91 (2009)
17. Chen, D., Nixon, M., Lin, T.: Over the Air Provisioning of Industrial Wireless Devices Using Elliptic Curve Cryptography. In: IEEE International Conference on Computer Science and Automation Engineering, CSAE (2011)
18. Gnanasivam, P.: Ear and Fingerprint Biometrics for Personal Identification. In: International Conference on Signal Processing, Communication, Computing and Networking Technologies, ICSCCN 2011 (2011)
19. Huang, Y., Ao, X., Li, Y.: Multiple Biometrics System based on DavinCi Platform. In: International Symposium on Information Science and Engineering, ISISE (2008)
20. Zhang, Y., Sun, D., Qiu, Z.: Hand-Based Feature Level Fusion for Single Sample Biometrics Recognition. In: International Workshop on Emerging Techniques and Challenges for Hand-Based Biometrics, ETCHB (2010)
21. Nilsson, D., Sun, L., Nakajima, T.: A Framework for Self-Verification of Firmware Updates over the Air in Vehicle ECUs. In: IEEE GLOBECOM Workshops (2008)
22. Guo, X., Huang, S., Nazhandali, L.: Fair and Comprehensive Performance Evaluation of 14 Second Round SHA-3 ASIC Implementations. In: NIST 2nd SHA-3 Candidate Conference (2010)
23. iOS OTA versioning data, XML format. Apple Inc. (2013), `http://mesu.apple.com/assets/com_apple_MobileAsset_SoftwareUpdate/com_apple_MobileAsset_SoftwareUpdate.xml`
24. Euler, L.: Theorematum quorundam ad numeros primos spectantium demonstratio. Commentarii Academiae Scientiarum Petropolitanae 8, 141–146 (1741)
25. National Institute of Standards and Technology: Recommended elliptic curves for federal government use (1999), `http://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTReCur.pdf`

Machine Learning Techniques for Anomalies Detection and Classification

Amira Sayed Abdel-Aziz[1], Aboul Ella Hassanien[2], Ahmad Taher Azar[3], and Sanaa El-Ola Hanafi[2]

[1] Université Française d'Égypte, Cairo, Egypt
amiraabdelaziz@gmail.com
[2] Faculty of Computers and Information, Cairo University, Egypt
aboitcairo@gmail.com
[3] Faculty of Computers and Information, Benha University, Egypt
ahmad_t_azar@yahoo.com

Abstract. Malicious users are always trying to intrude the information systems, taking advantage of different system vulnerabilities. As the Internet grows, the security limitations are becoming more crucial, facing such threats. Intrusion Detection Systems (IDS) are a common protecting systems that is used to detect malicious activity from inside and outside users of a system. It is very important to increase detection accuracy rate as possible, and get more information about the detected attacks, as one of the drawbacks of an anomaly IDS is the lack of detected attacks information. In this paper, an IDS is built using Genetic Algorithms (GA) and Principal Component Analysis (PCA) for feature selection, then some classification techniques are applied on the detected anomalies to define their classes. The results show that J48 mostly give better results than other classifiers, but for certain attacks Naive Bayes give the best results.

1 Introduction

The modern world has become very dependent on the cyber domain and web services as a major source of information and business. Hence, the need for network security is in increase day by day for secure transformation of information and protection of daily transactions and softwares used. As attacks get more sophisticated and advanced, intelligent security solutions need to emerge to face them. Intrusion Detection Systems (IDS) [1] [2] are commonly used to defend systems against anomalous activities from authorized and unauthorized users, where they can be placed on a host or inside a network. Many methodologies have been employed to implement IDSs, which can be basically categorized into Misuse and Anomaly based IDSs. They are usually implemented to be quick to train, and accurate to identify anomalies with law false alarms as possible.

A misuse-based IDS compares monitored connections to a database of attack signatures that has been built previously. The pros of such technique is that it is very accurate detecting and defining an attack and it has very law false alarm

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 219–229, 2013.

rate. The cons are as the database gets bigger, it tends to become slower; besides it fails to detect previously unknown attacks and the database has to be updated from time to time with new attacks signatures. An anomaly-based IDS builds a model that represents the normal behaviour of the system, and considers any deviation — within a certain degree — as anomalous or suspicious activity. Its pros is that it is capable of detecting new and unknown attacks, and it usually uses a learning method to adjust the normal model from time to time. Its cons are that it is incapable of defining which type of attack has been detected, it is defined as an anomaly only. It also has a high rate of false alarms as some of the normal activities may be detected as anomalous. Sometimes, a hybrid of both techniques may be applied. IDSs can be categorized also as: Host-based and Network-based (based on placement), Passive or Active (based on reaction), and Centralized and Distributed (based on structure).

A network-based IDS (NIDS) is employed in this paper, and it consists of 2 phases. The first phase is to classify the connections into normal and anomaly ones. The second stage involves the usage of classifiers to label the detected anomalies with their right class or find out it was originally normal but wrongfully classified as anomaly. Classifiers are a set of tools that are used to classify some given data into their correct classes, on the basis of some analysis and learning using previously labelled data. Different machine learning classifiers were used to classify and label the anomalies detected by a previously implemented NIDS [3], using features selected by Principal Component Analysis (PCA) technique. A comparison between the classification results were shown by the end with the analysis and conclusion. The paper is organized as: section 2 gives a background of basic concepts applied in the paper, section 3 describes the experiment and its settings, and section 4 for the conclusion.

2 Background

2.1 Anomaly Intrusion Detection

Anomaly intrusion detection approach is one of two approaches to be followed to build an IDS (the other is misuse intrusion detection). The main idea is to build a normal model of the system behaviour, and consider any deviation of that model as anomalous or suspicious. Anomaly detection has been an active research area, as different algorithms are investigated for best results. The main advantage of such technique is that it can detect new attacks and intrusions, without requiring any expert knowledge. Its disadvantages is that they have a high rate of false alarms, and it also can not provide details about the detected attacks [4] [5].

2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a technique that is usually used for classification and compression, as it reduces the data set dimensionality by extracting

a new feature set that is smaller than the original one. The new extracted feature set includes most of the sample data information, that is the present variation given by the correlations between the original variables. PCA helps identifying the patterns in data in a way that highlights their similarity and differences, by the feature reduction process. The new features — called Principal Components (PCs) — are ordered by the amount of total information retained, and they are uncorrelated [6] [7]. PCA process is shown in Figure 1 below.

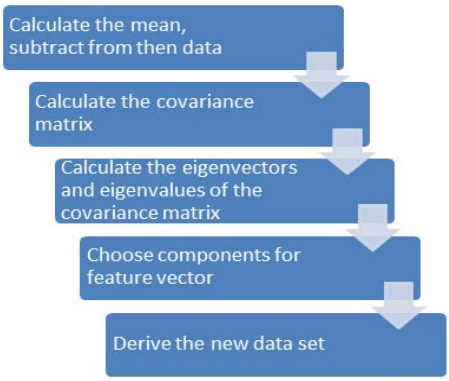


Fig. 1. The Principal Components Analysis Feature Reduction Process

2.3 Machine Learning for Classification

Analysis tools needed to find patterns and relationships in large data sets are known as data mining techniques. They could be statistical, machine learning, or mathematical models. Classification techniques are becoming more popular as they are used widely to process data, by analysing and categorizing them into known classes. Machine learning (ML) [8] [9] is a branch of Artificial Intelligence (AI),and its techniques are basically about helping systems to learn. ML techniques are applied to give IDSs generalization capabilities using limited training data. Since it is desirable in anomaly detectors to know and explain the reason for the anomalous activities to be able to find the right response, ML classifiers are useful in such case. Examples of ML techniques are: Decision Tree learning, Association Rule learning, Artificial Neural Networks, Genetic Programming, Support Vector Machines, Bayesian Networks, and many more.

In anomaly NIDS, traffic is usually classified into either normal or a specific attack category. Hence, a multi-category classifier is needed for such type of classification. Multi-category classifiers are either direct or indirect. Direct classifiers generally extend binary classifiers to deal with multi-category classification problems, while indirect classifiers decomposes the multi-category problem into multiple binary classification problems. For indirect classifiers, a base classifier is used to train the binary classification problems set, and results are merged using a combining strategy which works on the collected results of the binary classifiers [10] [11]. Figure 2 shows the process of data mining classification methods [12]

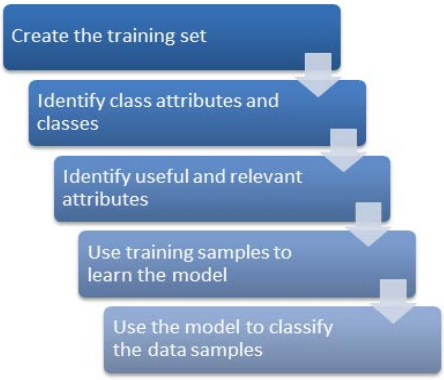


Fig. 2. The Process of Building a Classification Model

The following subsections explain the classifiers that were used in the experiment for anomalous activities classification.

Naive Bayes. Naive Bayes classifiers use the Bayes theorem, where each instance is described as a vector of attribute values, and it belongs to the class with the maximum posterior probability. They are fast and easy to interpret, but their limitation is the requirement of prior probability. They assume the attributes are independent, which may give satisfactory results but also might be a problem in its performance results sometimes, and they have lower performance on large databases [13] [14].

Decision Trees. A Decision Tree (DT) [15] [16] is a structure of layered nodes (a hierarchical organization of rules), where a non-terminal node represents a decision on a particular data item and a leaf (terminal) node represents a class. In this paper three types of decision trees were tested: J48 (C4.5) decision tree, Naive Bayes Tree (NBTree), and Best-First Tree (BFTree).

The C4.5 tree — or J48 as called in Weka tool — is based on the IDS algorithm, which tried to find simple decision trees. NBTree uses a high-mapping function to provide a simple and compact way to index high-dimensional data. It uses the Euclidean norm value as the index key for high-dimensional points, so that they can be ordered and used for searching. In BFTree, the nodes are expanded in the best-first order, where the best split node is added to the tree in each step. It results in the same fully-grown tree as the standard tree model, but it enables researchers to use cross-validation while pruning the tree to select multiple expansions [17].

Multilayer Perceptron Neural Network. Complex tasks can not be solved by single neurons in neural networks, and building the neural network by hand is very expensive. The perceptrons can only classify sets that are linearly separable,

and if the instances can not be separated linearly then the learning process will never be able to classify the instances properly. The solution is to use Multilayer Perceptron (MLP) (Werbos 1974, Rumelhart, McClelland, Hinton 1986), which is also known as feed forward networks. An MLP is a finite acyclic graph where neurons of the i-th layer serve as input features for neurons of i+1-th layer. An MLP can have multiple output neurons, based on the target values of the training patterns are described. Most MLPs have a connection structure, where all neurons of one layer are connected to all neurons of the next layer without shortcuts. All connections are weighted, and the hidden and the output neurons have a bias weight [18].

3 Experimental Results and Discussion

In this paper, the experiment is all about detecting intrusions and classifying the detected anomalies into their right class. As shown in figure 3, the process starts by discretizing categorical, real, and integral-valued features. Then, PCA is applied for features selection. After that – using the selected features – the GADG algorithm is applied to generate detectors for the intrusion detection process. Finally, the records detected as anomaly are lunched into the classifier to label them with proper attack class label.

The experiment was run using the NSL-KDD IDS benchmark data set [19] which contains four types of attacks: DoS (Denial of Service), Probe, U2R (User to Remote), and R2L (Remote to Local), beside the normal connections data. Table 1 shows the distribution of normal data and attacks in the Train set.

Table 1. Distributions of Attacks and Normal NSL-KDD train records

	Total Records	Normal	DoS	Probe	U2R	R2L
Train_20%	25192	13449	9234	2289	11	209
		53.39%	36.65%	9.09%	0.04%	0.83%
Train_All	125973	67343	45927	11656	52	995
		53.46%	36.46%	9.25%	0.04%	0.79%

3.1 Detectors Generation

The GADG (Genetic Algorithm for Detectors Generation) was originally suggested and applied in [20] [3], where The algorithm sequence is shown in figure 4. Equal-Width Binning was used for the discretization step,where the number of bins for each feature calculated using the following equation

$$k = max(1, 2 * \log l) \tag{1}$$

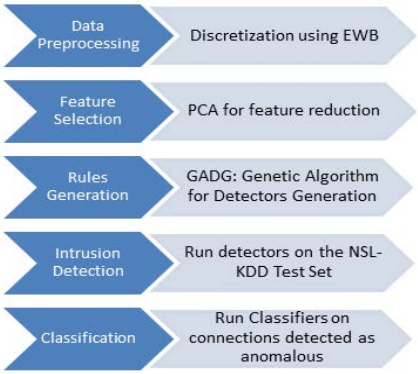


Fig. 3. The Intrusion Detection System Process

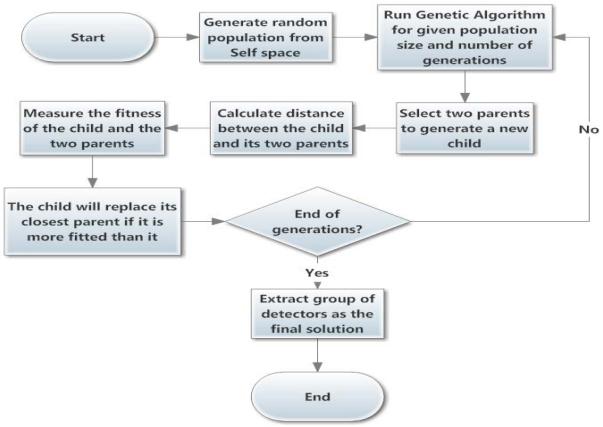


Fig. 4. The Genetic Algorithm Detectors Generation Process

3.2 Anomalies Classification Process

After the discrimination process done using the generated detectors, we should have the connections classified to either normal or anomaly. The anomaly connections are then lunched into the classifiers to either confirm it is an attack and get more information about the attack type, or to find it is not an attack and it is a normal connection.

3.3 Results

The PCA leads us to 22 features selected for best classification of connections, and they are mentioned in table 2.

As for the discrimination phase, the anomalies were detected using the detectors generated by the GA using the Euclidean distance. The results of this

Table 2. NSL-KDD Data Set Features Selected by PCA

Feature	Description	Type
1. duration	Duration of the connection	Integer
2. protocol type	Connection protocol (e.g. tcp, udp)	Categorical
3. service	Destination service (e.g. telnet, ftp)	Categorical
4. flag	Status flag of the connection	Categorical
5. source bytes	Bytes sent from source to destination	Integer
6. destination bytes	Bytes sent from destination to source	Integer
7. land	1 if connection is from/to the same host/port; 0 otherwise	Binary
8. wrong fragment	number of wrong fragments	Integer
9. urgent	number of urgent packets	Integer
11. failed logins	number of failed logins	Integer
13. num compromised	number of "compromised" conditions	Integer
14. root shell	1 if root shell is obtained; 0 otherwise	Binary
17. num file creations	number of file creation operations	Integer
18. num shells	number of shell prompts	Integer
22. is guest login	1 if the login is a "guest" login; 0 otherwise	Binary
27. rerror rate	% of connections that have "REJ" errors	Real
28. srv rerror rate	% of connections that have "REJ" errors	Real
29. same srv rate	% of connections to the same service	Real
31. srv diff host rate	% of connections to different hosts	Real
32. dst host count	count of connections having the same destination host	Integer
35. dst host diff srv rate	% of different services on the current host	Real
37. dst host srv diff host rate	% of connections to the same service coming from different hosts	Real

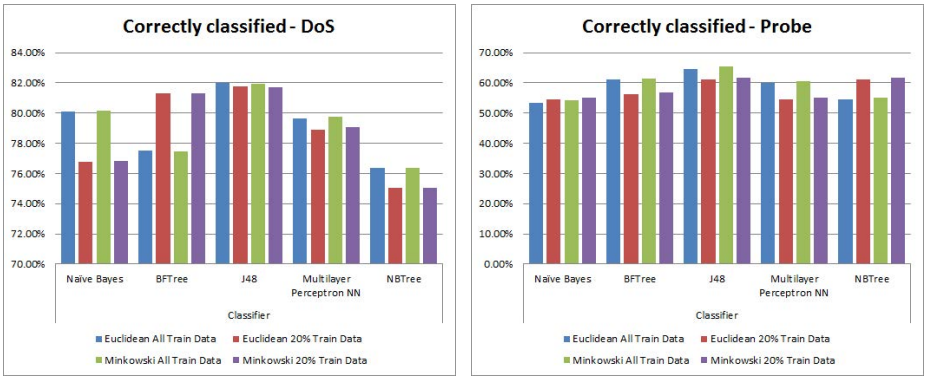
process were shown in [21]. The best detection results obtained using the features selected by PCA were the ones generated using population size 200 for number of generation 200, applying the Euclidean distance measure, and population size 200 for number of generations 500, applying the Minkowski distance measure in the GADG algorithm. Below are the results of the classification process using the classifiers described above. The classifiers were trained one time using 20%

of the training set, and another time using the whole training set. Table 3 and charts figure 5 to figure 7) show the statistics of the classification results of both models of each classifier.

Table 3. Results of Classification Process

	Train Data	NB	BFTree	J48	MLP	NBTree
Euclidean	All	63.49%	65.41%	70.07%	66.20%	65.25%
	20%	62.33%	66.99%	68.91%	66.74%	65.49%
Minkowski	All	65.90%	68.28%	72.88%	69.12%	67.01%
	20%	64.81%	69.54%	71.96%	69.82%	67.41%

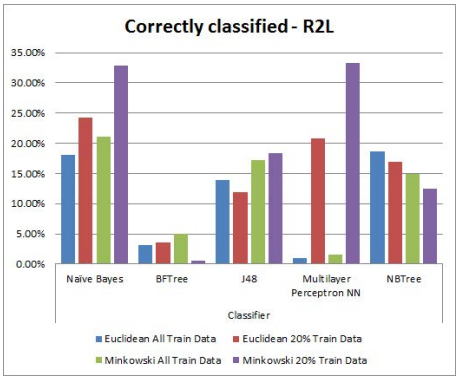
Looking into table 3, one can realize that as an overall, the J48 classifier give the best results, then the MLP, followed by BFTree. Viewing the results from the detectors point of view, the anomalies detected using the Minkowski distance are better classified results than those detected using the Euclidean distance. Based on the training set, sometimes using the whole train set give better results, in other times using 20% of the train set is enough for the classifier to give better results.



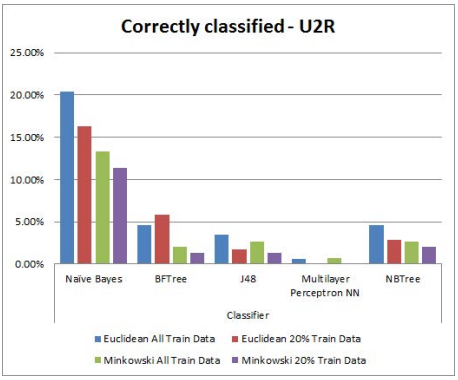
(a) The DoS Classification Results (b) The Probe Classification Results

Fig. 5. DoS and Probe Attacks Classification Results

In figure 5(a), the results of the DoS (Denial of Service) attack classification show that the J48 give the best results, using all the data in the train set, while the BFTree give close results using only 20% of the train data to build the model. Surprisingly, the NB classifier give close enough results of the correctly classified attacks, using all the train data to build the model. As for the Probe attack results shown in figure 5(b), again the J48 gives the best classification



(a) The R2L Classification Results



(b) The U2R Classification Results

Fig. 6. R2L and U2R Attacks Classification Results

Fig. 7. The Normal Classification Results

results using the whole train set, while following is the BFTree using the whole train set and the NBTree using 20% of the train set data. Figures 6(a) and 6(b) show the classification results of R2L and U2R attacks (respectively), and due to their low representation in the data set the results are not very high. For the R2L classification, the NB classifier in general give the best results, followed by the MLP with Minkowski anomalies, using 20% of the train set to build the model. The NB succeeds again to correctly classify 20% of the U2R attacks while other classifiers almost fails to do so. Finally for the normal data that were correctly detected as anomalies, it can be realized that all classifiers except the NB were successful to correctly classify 80% and more of the normal data, with the NBTree giving the best results that others.

4 Conclusion

From the results shown above, we can come up with few notifications. In general, decision trees give the best results.The NB classifier give the best results with the attacks that are least presented in the data set or have very few training records. It also give the best results in most cases when 20% of the train data is used to build the model. NBTree and J48 give better results in most cases if the whole train set is used to build the classification model, and BFTree give better results in most cases when the model is built using 20% of the training data. Finally, the MLP mostly give better results when the whole training set is used to build he classification model. As for the results of the first stage of detection, the anomalies detected using the Minkowski distance measurement are better classified than the anomalies detected using the Euclidean distance measurement.

So what do we come up with from these notifications? When 20% of the train set is used to build the classification model, obviously it takes less time for it to be built. Hence, if it gives the best results or even close to the best then it is better to use this portion of the train set, especially when time is a critical issue. A classifier very basic and fast- such as the NB can classify the data less presented in the training records, so it should be used in such cases — no need for any complex classifiers. Decision Trees have the advantage of being able to deal with different types of attributes, with transparency of knowledge, and are fast classifiers — hence, they give best results when data is well represented in the training set. Although neural networks (the MLP in the experiment) are fast classifiers and tolerant to highly interdependent attributes, they are not able to deal well with discrete values, hence the worse results in most cases. The future work will include a hierarchical model, where a multi-layer classifier is used for different types of attacks.

References

1. Murali, A., Roa, M.: A survey on intrusion detection approaches. In: First International Conference on Information and Communication Technologies, ICICT, pp. 233–240 (2005)
2. Garcia-Teodora, P., Díaz-Verdejo, J., Maciá–Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security 28(1-2), 18–28 (2009)
3. Aziz, A.S.A., Azar, A.T., Hassanien, A.E., Hanafi, S.E.O.: Continuous Features Discretizaion for Anomaly Intrusion Detectors Generation. In: WSC17 2012 Online Conference on Soft Computing in Industrial Applications (2012)
4. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In: Proceedings of the Third SIAM International Conference on Data Mining, vol. 3, pp. 25–36. SIAM (2003)
5. Brown, D.J., Suckow, B., Wang, T.: A Survey of Intrusion Detection Systems. TU Vienna, Austria (2000)
6. Jolliffe, I.T.: Principal component analysis, p. 487. Springer, New York (1986)

7. Lindsay, I.S.: A tutorial on principal components analysis. Cornell University, Ithaca (2002)
8. Tang, D.H., Cao, Z.: Machine Learning-based Intrusion Detection Algorithms. Journal of Computational Information Systems 5(6), 1825–1831 (2009)
9. Tran, T.P., Tsai, P., Jan, T., He, X.: Machine Learning Techniques for Network Intrusion Detection. Dynamic and Advanced Data Mining for Progressing Technological Development: Innovations and Systemic Approaches (2010)
10. Khoshgoftaar, T.M., Gao, K., Ibrahim, N.H.: Evaluating indirect and direct classification techniques for network intrusion detection. Intelligent Data Analysis 9(3), 309–326 (2005)
11. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. Informatica 31, 249–268 (2007)
12. Joshi, M.: Classification, Clustering, and Intrusion Detection Systems. International Journal of Engineering Research and Applications (IHERA) 2(2), 961–964 (2012)
13. Zhang, H.: The optimality of naive Bayes. In: Proceedings of the FLAIRS Conference, vol. 1(2), pp. 3–9 (2004)
14. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 161–168. ACM (2006)
15. Kruegel, C., Tóth, T.: Using decision trees to improve signature-based intrusion detection. In: Vigna, G., Kruegel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 173–191. Springer, Heidelberg (2003)
16. Mitchell, T.M.: Machine learning. McGraw Hill, Burr Ridge (1997)
17. Shi, H.: Best-first decision tree learning. PhD dissertation, The University of Waikato (2007)
18. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine learning, neural and statistical classification (1994)
19. NSL-KDD Intrusion Detection data set, `http://iscx.ca/NSL-KDD/`
20. Aziz, A.S.A., Salama, M.A., Hassanien, A.E., Hanafi, S.E.O.: Artificial Immune System Inspired Intrusion Detection System Using Genetic Algorithm. In: Chojnacki, A. (Guest ed.): Special Issue: Advances in Network Systems, vol. 36, pp. 347–357 (2012)
21. Aziz, A.S.A., Azar, A.T., Hassanien, A.E., Hanafy, S.E.O.: Genetic Algorithm with Different Feature Selection Techniques for Anomaly Detectors Generation. In: Federated Conference on Computer Science and Information Systems (FedCSIS 2013). IEEE (submitted, 2013)

Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing

Nor Fatimah Awang and Azizah Abd Manaf

Advanced Informatics School (UTM AIS),
UTM International Campus,
Kuala Lumpur, Malaysia
norfatimah@upnm.edu.my, azizah07@ic.utm.my

Abstract. Today, web applications are becoming the most popular tool that offers a collection of various services to users. However, previous research and study showed that many web applications are deployed with critical vulnerabilities. Penetration testing is one of the well-known techniques that is frequently used for the detection of security vulnerabilities in web application. This technique can be performed either manually or by using automated tools. However, according to previous study, automated black box tools have detected more vulnerability with high false positive rate. Therefore, this paper proposed a framework which combines both automated black box testing and manual penetration testing to achieve the accuracy in vulnerability detecting in web application.

1 Introduction

Today, Internet has become the most ever powerful tool for user throughout the world. The internet offers a collection of various services and resources, not only email and World Wide Web as the principle constituents of internet. According to the World Internet Usage and Population Statistics as of June 2012, the number of current Internet user has raised to 2.4 billion, which is estimated to be equal to 34.3% of the world population [1]. This number is a 566.4% increase since the year 2000. Due to the simplicity of its use and its high accessibility, the Web has become the dominant way for people to search information, online banking, job seeking, purchasing tickets, hotel reservations and social networking. Nowadays, popular networking sites such as MySpace and Facebook, with exciting and interactive user driven content such as blog and YouTube videos, are becoming the norm for web content. The growth of these sites gave a high impact and business opportunity to the organization. Several recent studies [2], [3] indicate that this popularity of web applications has unfortunately also attracted attention of attackers. This is reflected by various statistics. For example, Fig. 1, taken from the 2012 IBM Internet Security X-Force 2012 Trend and Risk Report, presents web application vulnerabilities versus total vulnerabilities [2]. It shows the increase in the number of web application vulnerability from 2011 to 2012. Web application vulnerabilities increase to 14%

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 230–239, 2013.

from 2,921 vulnerabilities in 2011 to 3,551 vulnerabilities in 2012. Similarly, another study conducted by Positive Technology in 2010 and 2011, reported that from 123 websites chosen for study, all the sites contained vulnerabilities whereby 60% of sites are vulnerable to high-risk, 98% medium and 37% are vulnerable to low-risk [3]. All the above studies show the potential serious vulnerabilities which exist in web applications.

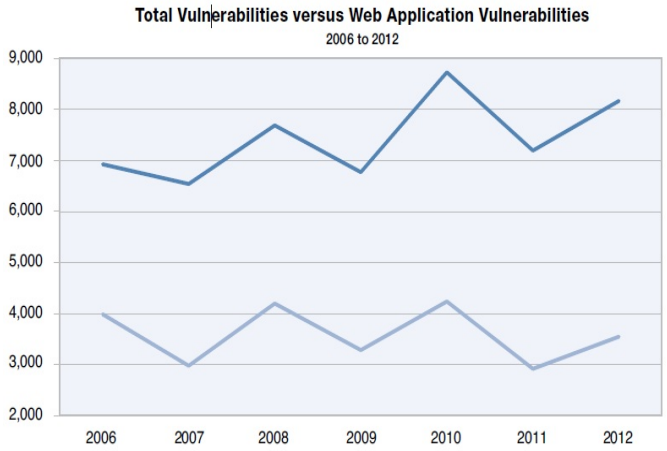


Fig. 1. Total Vulnerabilities versus Web Application [2]

In information security, there are no accepted standard regarding the word vulnerability. We are trying to define the word vulnerability related to web application. There are many definitions proposed, here are some of them:

- According to Wang et al., vulnerability is security flaw, defect or mistake in software that can be directly used by a hacker to gain access to system or network [4].
- Weakness in the security system which might be exploited by malicious users causing loss or harm [5].

Here we can consider the term vulnerability is applied to one weakness in a system, which allows an attacker to violate the integrity of that system. Based on the above scenario, security is a big issue that should be seriously considered by the system administrator as well as top management in order to protect the potential software and web application systems.

The main objective of this paper is to propose a framework for detecting vulnerability in web applications using both techniques automated black box and manual penetration testing.

The structure of this paper is as follows. Section 2 briefly describes background of web application vulnerability and discusses on more related techniques that are commonly used to detect the vulnerabilities which exist in web application. Section 3

describes more detail on the proposed framework. Section 4 discusses the results and finally, section 5 presents the conclusion.

2 Background and Related Work

In order to understand the variety of vulnerabilities in web application, this section will give some explanations on the architecture and processes of web application, web application vulnerability and several techniques that can be used to discover vulnerability.

2.1 Web Application Architecture

Web application architecture is often structured as a three-tiered application [6],[7]. The architecture of web applications consists of web browser, web server, web application and database server. Tier 1 architecture consists of web browser and web server, Tier 2 for web application and Tier 3 for database.

- Tier 1 – web browser and web server architecture

  A web browser is also known as web client, functions as the user interface to web server to get input from web application or database server. Web server receives input and interacts with client through web browser by using Hyper Text Transfer Protocol (HTTP) or via secure protocol HTTPS. There are many type of web servers where Apache and Internet Information Server (IIS) are the most popular web server in the world.

- Tier 2 – web application architecture

  Web application consists of a collection of scripts such as Javascript, VBscript, which reside on a web server and interact with databases or other sources of dynamic content. The common example, the data input using a web browser is processed and stored into database. Java Server Pages (JSP), PHP, Active Server Pages (ASP), Perl and Common Gateway Interface (CGI) are among the technology used to build web based application. Using the infrastructure of the Internet, web applications allow service providers and clients to share and manipulate information in a platform-independent manner. Normally web application server is attached on top of web server and works as interface from web client and database server. Web server will manage the page requested from web client by sending to application server and application server constructs code dynamically and passed back to web server. The flow of data among tiers gives rise to the input validation problem for the web application server; it must check and/or modify incoming input before processing them further or incorporating them into output that it passes to other tiers to execute. Failure to check or sanitize input appropriately can compromise the web application's security [8].

- Tier 3 – database architecture

  Stores and manages all the processed users input data. The database tier is responsible for the access of authenticated users and the rejection of malicious users from the database.

2.2 Web Application Vulnerability

Because web applications are open to the world, they are more vulnerable to attacks and prone to a great variety of vulnerabilities. In this section, we describe some of the most common and well-known web application vulnerabilities based on OWASP Top Ten lists 2010 [9]. OWASP stands for Open Web Applications Security Project, and is an open-source collaboration of web based security tools, technologies and methodologies from industry leaders, educational organizations and individuals from around the world. The OWASP Top Ten is a valuable document for developers and testers because its focus on web applications. The OWASP Top 10 2010 has listed the ten most critical web application security vulnerabilities as shown in Table 1. The OWASP Top 10 2010 refers to the top 10 web attacks as seen over the year by security experts, and community contributors to the project.

Table 1. OWASP Top Ten Vulnerability 2010 [9]

Ranking	Vulnerability
A1	Injection
A2	Cross site scripting (XSS)
A3	Broken Authentication / Session Management
A4	Insecure Direct Object References
A5	Cross Site Request Forgery
A6	Security Misconfiguration Sensitive Data Exposure
A7	Insecure Cryptographic Storage
A8	Failure to Restrict URL Access
A9	Insufficient Transport Layer Protection
A10	Unvalidated Redirects and Forwards

Injection and Cross Site Scripting attacks are the most common popular vulnerabilities exploited by attackers. In this paper, we focus on SQL Injection and Cross Site Scripting due to the common vulnerabilities that have evolved in the last decade [10]. These attacks are triggered where an attacker intentionally sends a malicious input or script to the application to get some valuable information. The detail attacks are as follows [4]:

- SQL Injection Attack

  This kind of attack occurs when an attacker uses some special SQL queries as an input, which can open up a database [11], [12], [13]. Online forms such as login prompts, search enquiries, guest books and feedback forms are always targeted. The simple test to check for SQL injection attack is to append "or+1=1" to the URL and shows the data returned by the server.

- Cross Site Scripting (XSS) Attack

  Many XSS attacks happen because vulnerable applications fail to sanitize malicious input at either server side or browser side, allowing them to be injected into response pages. XSS attacks exploit through inputs that might contain HTML tags and Java Script code [14], [15]. For example, an attacker injects a malicious JavaScript program into a trusted site's web page. If a victim user visits the affected web page, then the web browser executes the malicious JavaScript program as though it came from the trusted site. By using this vulnerability, an attacker can force a client, such as a user web browser, to execute attacker-supplied code. As a result, the attacker's code is granted access critical information that was issued by the trusted site.

2.3 Techniques to Detect Vulnerability

Penetration testing and code analysis are two (2) well-known techniques frequently used by web developers and testers for detecting vulnerability [16],[17]. Penetration testing involves in stressing the application from the point of view of an attacker by using specific malicious inputs. This technique can be performed either manually or by using automated tools. Automated tool is also known as automated black box tools and most of these automated black box tools are commercial tools such as IBM Rational AppScan [19], Acunetix Web Vulnerability Scanner [20] and HP WebInspect [21] whereas, static code analysis is a technique that analyzes the source code of the application without trying to execute it, whilst searching for potential vulnerabilities. Similarly with penetration testing, this technique can be done manually or by using code analysis tools like Pixy [22], FORTIFY [23] or Ounce [24]. According to OWASP [4], the most efficient way of finding vulnerabilities in web application is manual code analysis. This technique is very time consuming. It requires expert skills and is prone to be overlooked error of a system. Therefore, due to time constraints or resource limitations, developers or testers frequently have to choose automated tools to search for vulnerability [18]. However, according to previous study conducted by [17], [18], automated black box tools have detected more vulnerabilities with high false positive rate (vulnerability detected that did not exist in web application). Therefore, this paper presents a framework that combines both automated black box testing and manual penetration testing to achieve better accuracy of vulnerability detecting in web application.

3 The Proposed Framework

Our proposed framework consists of four phases, refer to Fig. 2:

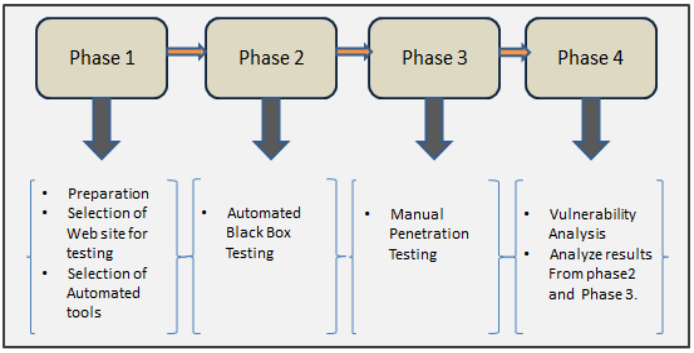


Fig. 2. Framework for detecting vulnerability in web application

- Phase 1

  The Web Application Security Scanner Evaluation Criteria (WASSEC) has set a guideline to evaluate web application scanners on their ability to effectively test web applications and identify vulnerabilities [25]. Other studies for comparing the effectiveness of web application scanning tools conducted by researchers [26, 27], was also used as a guideline for selecting the scanning tools. In this framework, we have considered and selected well known commercial automated black box tools namely IBM Rational AppScan [19] and we have chosen an anonymous online web application website as shown detail in Fig. 3.

Target IP/Hostname	Operating System	Web Server	Web Application
www.[redacted]	Unix	Apache 2.2.3 Redhat	PHP

Fig. 3. Target Testing Website

- Phase 2

  This phase uses the automated black box tool to scan the services to identify potential vulnerabilities in web application. The tool used in this phase in order to scan and detect the variety of known vulnerabilities in web application as listed in Table 1. To begin a scanning session, the tester must enter the entry URL of the web application. The tester then must specify options for the scanner's page crawler, in order to maximize page scanning coverage. In this phase, we always set the scanner to run in automated mode to maximize vulnerability detection capability.

- Phase 3

  This stage performs manual penetration testing to confirm vulnerabilities that have been detected through second phase (to check false positive of the vulnerabilities).

- Phase 4

  The goal of this phase is to analyze the result of the target system after conducting testing. Vulnerability analysis result will be based on two activity's results from two different phases, phase 2 and phase 3. This activity will conclude and validate the vulnerability whether the vulnerability reported in phase 2 is actual vulnerability and to ensure no false positive exist in the test result.

4 Results and Discussion

This section presents the results that have been performed as described in section 3. The testing was done in actual anonymous online web application as shown detail in Fig. 3. As you can see in Fig. 4, a total of 27 vulnerabilities have been detected by automated black box tools of which the medium and high severity level share the same amount with the remainder categorized as low. Meanwhile, Fig. 5 shows five different types of vulnerabilities namely:

1) SQL Injection : it is possible to alter and steal the information stored in database
2) Content Spoofing: it is possible to trick a user to believe that certain content appearing on a Web site is legitimate and not from an external source.
3) Directory Indexing: it is possible to allow the contents of unintended directory listings to be disclosed to the user
4) Information Leakage: it is possible to reveal sensitive information, such as from developer comments or error messages
5) Abuse of functionality: it is possible to use a web site's own features and functionality to attack itself or others.

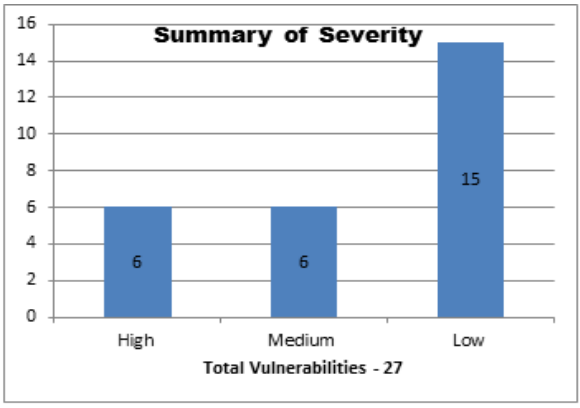


Fig. 4. Number of Vulnerabilities by severity level

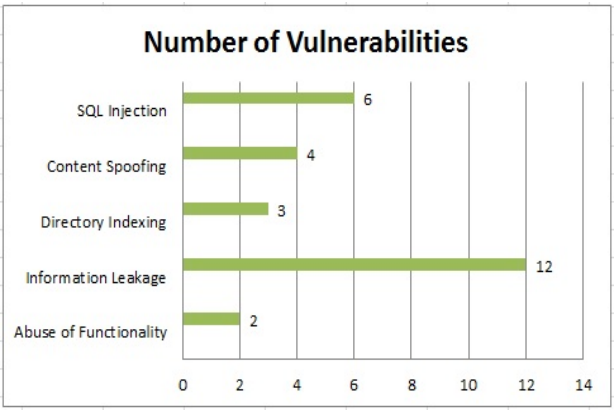


Fig. 5. Number of Vulnerabilities by type

The result shows that SQL Injection is classified as high severity; content spoofing and abuse of functionality are categorized as medium severity. Meanwhile, information leakage and directory indexing are categorized as low severity.

Possible vulnerabilities that have been reported in Fig. 5, is validated again by using manual penetration testing to ensure that false positive does not exist in this testing phase. We consider that vulnerability exists, if any malicious patterns or errors were found as shown detail in Fig. 6. This work is still in progress to validate all vulnerabilities detected by automated tool. In this paper, we have chosen SQL Injection vulnerabilities due to high severity level to validate whether false positive or not. As you can see in Table 2, results so far do not have false positives. It means that the potential vulnerability detected by automated black box tool is considered as actual vulnerability after successfully being reconfirmed by manual penetration testing.

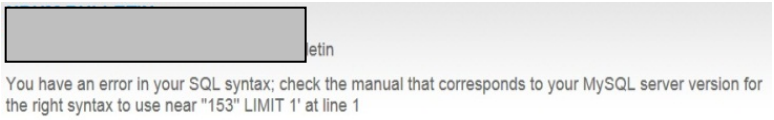


Fig. 6. Example of error after conducting manual testing

Table 2. Reconfirm False Positive

Type of Vulnerability	# of Vulnerability	# of False Positive
SQL Injection	6	0

5 Conclusion

This paper presents a framework for detecting vulnerability in web application. One automated black box tool was selected to detect various vulnerabilities in web application. In this work, the automated tool pointed out five different type vulnerabilities as shown in Fig. 5. After detecting vulnerability process was completed in phase 2, manual penetration testing was performed in order to ensure there are no false positive exist in the test result.

References

1. Internet World Stats, Usage and Population Statistics (2013), http://www.internetworldstats.com/stats.htm
2. X-Force Research and Development Team, IBM X-Force 2012 Trend and Risk Report, Technical Report (March 2012)
3. Web Application Vulnerability Statistics for 2011-2012, Positive Technology, Technical Report (2012)
4. Wang, J.A., Guo, M., Wang, H., Xia, M., Zhou, L.: Environmental metrics for software security based on a vulnerability ontology. In: Third IEEE International Conference on Secure Software Integration and Reliability Improvement, pp. 159–168 (2009)
5. Pfleeger, C.P., Pfleeger, S.L.: Security in Computing, 3rd edn. Prentice Hall PTR (2003)
6. Kim, J.: Injection Attack Detection Using the Removal of SQL Query Attribute Values. In: 2011 International Conference on Information Science and Applications, ICISA, April 26-29, pp. 1–7 (2011)
7. Zhendong, S., Wassermann, G.: The Essence of Command Injection Attacks in Web Applications. In: Conference Record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 372–382 (2006)
8. Shklar, L., Rosen, R.: Web Application Architecture: Principles, Protocols and Practices, 2nd edn. John Wiley & Sons (2009)
9. 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
10. Theodoor, S., Davide, B., Engin K.: Have things changed now? An Empirical Study on Input Validation Vulnerabilities in Web Applications (2012), http://iseclab.org/papers/theo-journal.pdf
11. Ezumalai, R., Aghila, G.: Combinatorial Approach for Preventing SQL Injection Attacks, Advance Computing Conference. IEEE International, IACC (2009)
12. Justin, C.: SQL Injection Attacks and Defense. Syngress Publishing (2009) ISBN 13: 978-1-59749-424-3
13. Huang, Y., Yu, F., Hang, C., Tsai, C.H., Lee, D.T., Kuo, S.Y.: Securing Web Application Code by Static Analysis and Runtime Protection. In: Proceedings of the 12th International World Wide Web Conference, WWW 2004 (May 2004)
14. Shahriar, H., Zulkernine, M.: Taxonomy and classification of automatic monitoring of program security vulnerability exploitations. Journal of Systems and Software 84, 250–269 (2011) ISSN 0164-1212, 10.1016/j.jss.2010.09.020
15. Avancini, A.: Security testing of web applications: A research plan. In: 2012 34th International Conference on Software Engineering, ICSE, June 2-9, pp. 1491–1494 (2012)

16. Bacudio, A.G., Yuan, X., Chu, B.B., Jones, M.: An Overview of Penetration Testing. International Journal of Network Security & Its Applications (IJNSA) (November 2011)
17. Vieira, M., Antunes, N., Madeira, H.: Using Web Security Scanners to Detect Vulnerabilities in Web Services. In: IEEE/IFIP Intl Conf. on Dependable Systems and Networks, DSN (2009)
18. Nuno, A., Marco, V.: Comparing of Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services. In: 15th IEEE Pacific Rim International Symposium on Dependable Computing (2009)
19. IBM Security Appscan, http://www-01.ibm.com/software/awdtools/appscan/
20. Acunetic, http://www.acunetix.com/
21. HP WebInspect, http://www8.hp.com/my/en/software-solutions/software.html?compURI=1341991
22. Jovanovic, N., Kruegel, C., Kirda, E.: Pixy: a static analysis tool for detecting Web application vulnerabilities. In: 2006 IEEE Symposium on Security and Privacy, May 21-24, p. 6 p. 263 (2006)
23. FORTIFY, http://www.fortifysoftware.com/
24. Ounce, http://www.ouncelabs.com/
25. Web Application Security Scanner Evaluation Criteria Version 1.0, http://projects.webappsec.org/w/page/13246986/Web%20Application%20Security%20Scanner%20Evaluation%20Criteria
26. Doupé, A., Cova, M., Vigna, G.: Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In: Kreibich, C., Jahnke, M. (eds.) DIMVA 2010. LNCS, vol. 6201, pp. 111–131. Springer, Heidelberg (2010)
27. Fonseca, J., Vieira, M., Madeira, H.: Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks. In: The 13th IEEE Pacific Rim International Symposium on Dependable Computing (December 2007)

Linear Correlation-Based Feature Selection for Network Intrusion Detection Model

Heba F. Eid[1,5], Aboul Ella Hassanien[2,5],
Tai-hoon Kim[3], and Soumya Banerjee[4,5]

[1] Faculty of Science, Al-Azhar University, Cairo, Egypt
heba.fathy@yahoo.com
[2] Faculty of Computers and Information, Cairo University, Egypt
aboitcairo@fci-cu.edu.eg
[3] Hannam University, Korea
taihoonn@empas.com
[4] Dept. of CS, Birla Institute of Technology, Mesra India
dr.soumya@ieee.org
[5] Scientific Research Group in Egypt (SRGE)
http://www.egyptscience.net

Abstract. Feature selection is a preprocessing phase to machine learning, which leads to increase the classification accuracy and reduce its complexity. However, the increase of data dimensionality poses a challenge to many existing feature selection methods. This paper formulates and validates a method for selecting optimal feature subset based on the analysis of the Pearson correlation coefficients. We adopt the correlation analysis between two variables as a feature goodness measure. Where, a feature is good if it is highly correlated to the class and is low correlated to the other features. To evaluate the proposed Feature selection method, experiments are applied on NSL-KDD dataset. The experiments shows that, the number of features is reduced from 41 to 17 features, which leads to improve the classification accuracy to 99.1%. Also,The efficiency of the proposed linear correlation feature selection method is demonstrated through extensive comparisons with other well known feature selection methods.

Keywords: Network security, Data Reduction, Feature selection, Linear Correlation, Intrusion detection.

1 Introduction

Intrusion detection system (IDS) dynamically identify unusual access or attacks to secure the network channels [1, 2]. Network-based IDS (NIDS) is a major research problem, since it is a valuable tool for the defense in depth of computer networks. NIDS search for known or potential malicious activities in the network traffic and raises an alarm whenever a suspicious activity is detected.

However, an important research challenge for constructing high performance NIDS is dealing with data containing large number of features. Redundant features of the dataset complex the NIDS and reduce the classification accuracy

A.I. Awad, A.E. Hassanien, and K. Baba (Eds.): SecNet 2013, CCIS 381, pp. 240–248, 2013.

as well. Therefor, data reduction is an active research area in the field of machine learning and pattern recognition [3–5]. The dimensionality reduction of the dataset can be achieved by feature selection (FS). FS methods select an optimal subset of features that are necessary to increase the classification accuracy and reduce the time of the learning process [6,7]. Different feature selection methods are proposed to enhance the performance of IDS [8,9]. To evaluate the feature selected subsets, a feature goodness measure is require. In general, a feature is good if is not relevant with other features and is relevant to the output classes. On of the most important goodness metrics to select the features is Pearson correlation coefficients [10].

This paper propose a Linear correlation-based feature selection approach for building NID model. The proposed approach aims to improve the network intrusion classification accuracy by reducing the data dimensionality. It consists of two layers. The first layer select a feature subset based on the analysis of Pearson correlation coefficients between the features. While, the second layer select a new set of features from within the first layer redacted features subset; by measuring the Pearson correlation coefficients between the selected features and the classes.

The rest of this paper is organized as follows: Section 2 gives an overview of data Pre-Processing Approaches: feature selection and Linear correlation. Section 3 describes the NSL-KDD network intrusion dataset. Section 4 presents the proposed model of the network intrusion detection system. The experimental results and conclusions are discussed in Section 5 and 6 respectively.

2 Data Preprocessing

2.1 Feature Selection

Data reduction is a preprocessing step for classification. It aims to improve the classification performance through the removal of redundant features. Data reduction can be achieved by feature selection (FS). FS approaches generate a new set of features by selecting only a subset of the original features.

FS methods fall into two categories: filter approach [6,11] and wrapper approach [12,13]. Filter approaches depends on the general characteristics of the data to select the new set of features. The features are ranked based on certain statistical criteria, where the features with highest ranking values are selected. Frequently used filter methods include Pearson correlation coefficients [14], chi-square [15] and information gain [16].

While, wrapper approaches use a predetermined machine algorithm to select the new features subset. Wrapper approaches use the classification performance as the evaluation criterion . Genetic algorithm (GA) [17], ID3 [18] and Bayesian networks [19] are commonly used as induction algorithm for wrapper approaches.

2.2 Linear Correlation

The linear Correlation is a well-known similarity measure between two random variables. Pearson correlation coefficient (ρ); the Linear correlation coefficien; is a measure of dependence between two random variables [20].

For a pair of variables X with values x_i and Y with values y_i , the Pearson correlation coefficient ρ is given by the equation:

$$\rho = \frac{cov(X,Y)}{\sqrt{\sigma^2(X)\sigma^2(Y)}} \tag{1}$$

where cov is the covariance and σ is the variance.

The estimation of the Pearson correlation coefficient ρ is given by:

$$\rho = \frac{E(XY) - E(X)E(Y)}{\sqrt{\sigma^2(X)\sigma^2(Y)}} \tag{2}$$

$$\rho = \frac{\sum_i (x_i - \overline{x}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_i (x_i - \overline{x}_i)^2 \sum_i (y_i - \overline{y}_i)^2}} \tag{3}$$

where $\overline{x}_i$ is the mean of X, and $\overline{y}_i$ is the mean of Y .

The value of ρ lies between -1 and 1, if X and Y are linearly dependent (correlated), and $\rho = 0$ if X and Y are totally independent (uncorrelated).

Thus, features redundancies can be detected by correlation analysis. Where, a feature which is strongly correlated to some other features is a redundant one.

3 Network Intrusion DataSet: The NSL-KDD

The NSL-KDD dataset [21] is a benchmark used for the evaluation of network intrusion detection systems. NSL-KDD consists of selected records of the complete KDD'99 dataset [22]. Where, each NSL-KDD connection record contains 41 features and is labeled as either normal or an attack. The NSL-KDD dataset contain a train set and a test set. The training set contains a total of 22 training attack types, and the testing set contains an additional 17 types of attacks. The attacks fall into four categories: (1)DoS e.g Neptune, Smurf, Pod and Teardrop, (2)R2L e.g Guess-password, Ftp-write, Imap and Phf, (3)U2R e.g Buffer-overflow, Load-module, Perl and Spy, and (4)Probing eg. Port-sweep, IP-sweep, Nmap and Satan. Table 1 gives a description of the first ten features of the NSL-KDD dataset.

4 Modeling the Linear Correlation Based FS for Network Intrusion Detection

The design of the proposed NID model is shown in Fig 1. It is comprised of the following three fundamental building layers: Layer(1) Feature selection by applying correlation analysis between the 41 features. Layer (2) Feature selection by applying correlation analysis between the selected features and the classes, and Layer(3) Intrusion detection and classification of a new intrusion into five outcome.

Table 1. NSL-KDD dataset Features Sample

Feature	Description
1. duration	Duration of the connection.
2. protocol type	Connection protocol (e.g. tcp, udp).
3. service	Destination service (e.g. telnet, ftp)
4. flag	Status flag of the connection
5. source bytes	Bytes sent from source to destination
6. destination bytes	Bytes sent from destination to source
7. land	1 if connection is from/to the same host/port; 0 otherwise
8. wrong fragment	number of wrong fragments
9. urgent	number of urgent packets
10. hot	number of "hot" indicators

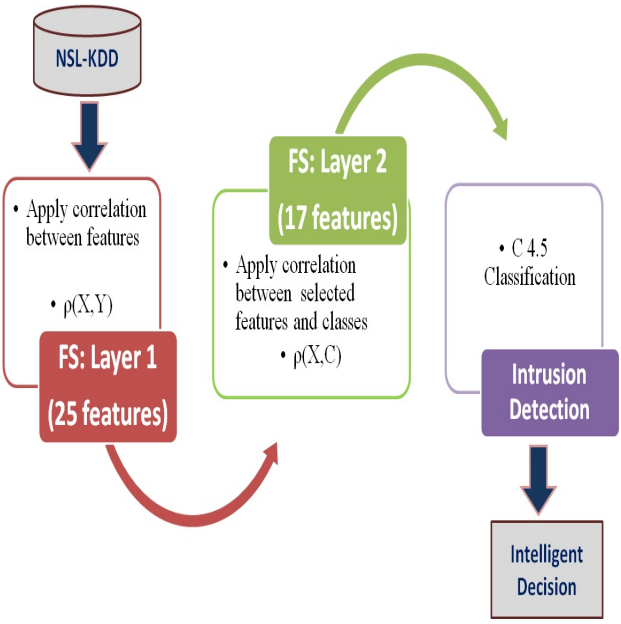


Fig. 1. The proposed linear correlation-based NID model

4.1 Layer 1: Finding the Correlation between the Features

Pearson correlation coefficient ρ is computed for each feature with the other 41 features of the NSL-KDD dataset according to equation 3; to form the $\rho(X, Y)$ matrix of the NSL-KDD 41 features. Then, for each feature the maximum value of ρ and its corresponding feature is located.

A features is highly correlated with other features as ρ go near to 1. Which means that, one feature contains lots of information about the other and implies

that knowing one feature can provide enough information the other feature can give. Thus, one of the feature is consider to be a redundant feature and can be deleted.

Following this concept, a fitness of $\rho > 0.1$ is assigned to rank the $max\rho$ features. Where, each feature at the $max\rho$ column that satisfy this fitness is selected.

4.2 Layer 2: Finding the Correlation between the Selected Features and the Classes

At layer 2, the features subset selected from layer 1 is reduced. The reduction is done based on calculating the Pearson Correlation coefficients between each selected feature and the classes (c_j). The Pearson correlation coefficients $\rho(X, C)$ is computed according to the following equation:

$$\rho = \frac{\sum_i (x_i - \overline{x}_i)(c_j - \overline{c}_j)}{\sqrt{\sum_i (x_i - \overline{x}_i)^2 \sum_j (c_i - \overline{c}_j)^2}}, j = 1, ..., 5 \quad (4)$$

4.3 Layer 3: Intrusion Classification

we evaluate the performance of the proposed linear correlated based FS for designing NIDS on C4.5 classifier. The C4.5 classifier classify the NSL-KDD dataset to five outcomes; normal and four types of attacks.

5 Experiments and Analysis

5.1 Evaluation Criteria

The Comparison Criteria to evaluate the proposed network intrusion detection system are: (1) the classification Accuracy and (2) the speed of the ID system.

Classification performance of ID system is measured in term of the $F - measure$; which is calculated based on the confusion matrix shown in Table 2. The F-measure is a weighted mean that assesses the trade-off between *precision* and *recall* . An ID system should achieve a high recall without loss of precision.

Table 2. Confusion Matrix

		Predicted Class	
		Normal	Attake
Actual Class	Normal	True positives (TP)	False negatives (FN)
	Attake	False positives (FP)	True negatives (TN)

True negatives (TN) as well as True positives (TP) correspond to a correct prediction of the that normal and attacks events. False positives (FP) refer to

normal events being predicted as attacks; while False negatives (FN) are attack events incorrectly predicted as normal [23].

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{7}$$

5.2 Results and Discussions

The proposed linear correlation- based NID model is evaluated using the NSL-KDD dataset, where 59586 records are randomly taken. All experiments have been performed using Intel Core 2 Duo 2.26 GHz processor with 2 GB of RAM.

Experiments 1: Evaluation of the Proposed Linear Correlation-Based Feature Selection Approach. The Pearson correlation coefficients matrix of the 41 features of the NSL-KDD data set are computed. 25 features are selected from the total 41 features, based on the analysis of the maximum $\rho(X, Y)$ between the features and the fitness $\rho > 0.1$. The selected 25 features from layer 1 are given in table 3.

Table 3. Selected features based on maximum ρ between features (25 features)

$\rho(X, Y)$; $\rho > 0.1$
22,8,36,26,25,2,19,27,34,16,18,13,10,24,23, 38,39,28,41,35,37,33,40,31,5,6.

Then, the Pearson correlation coefficients between the 25 selected features and the 5 classes are calculated. Which reduced the 25 features to 17 features as shown in table 4.

Table 4. Selected features based on ρ between 25 features and classes (17 features)

$\rho(X, C)$
22,36,26,25,27,18,23,38,39, 28,41,35,37,40,31,5,6.

Table 5 gives the F-measures for the C4.5 classifiers; with the full dimension of the NSL-KDD dataset (41 features) and after applying the proposed linear correlation based FS. The comparison results are based on 10 fold cross-validation.

Table 5. $F - Measure$ comparison of the proposed linear correlation-based feature selection approach

Feature selection approach	Number of features	F-Measure
Non	41	97.9%
proposed linear correlation-based	17	99.1%

It is clear from table 5 that for the proposed linear correlation-based feature selection the classification accuracy increased to 99.1% while the number of features are decreased to 17 features.

Table 6 gives the timing speed of building the proposed hybrid NID model; which hybrid the proposed linear correlation-based FS with C4.5 classifier.

Table 6. Timing and Testing accuracy comparison of linear correlation-based feature selection approach

	Time to build model (sec)	Test accuracy
C4.5	36.15	97.9%
Hybrid C4.5 with linear correlation-based	12.02	99.1 %

From Table 6 it is clear that the timing speed is improved to 12.02 second, which is very important if real time network applications is desired. Also, the classification accuracy achieved using the proposed FS approach is improved to 99.1%, than using a standalone C4.5 classifier.

Experiments 2: Proposed Linear Correlation-Based Feature Selection vs. Different Feature Selection Methods. Various well known feature selection approaches as PCA, Gain Ratio and information gain are compared with the proposed linear correlation-based feature selection approach. Table 7 gives the F-Measure accuracy of the reduced data using the linear correlation-based feature selection and the other well known feature selection methods. The F-Measure accuracy is based on 10 fold cross-validation

Table 7. $F - Measure$ comparison of the proposed linear correlation-based feature selection and other feature selection methods

Feature selection approach	Number of features	F-Measure
PCA	25	97.6%
Gain-Ratio	34	98.8%
Information Gain	35	98.6%
Linear correlation-based	17	99.1%

From table 7, it is clear that the F-measure for the linear correlation-based approach shows better result when compared to the other well known FS approaches.

6 Conclusion

In this paper, we propose a linear correlation-based feature selection method for building NID model. The proposed feature selection method introduce an efficient way of analyzing feature redundancy. It consists of two layers, where the first layer select a feature subset based on the analysis of Pearson correlation coefficients between the features. While, at the second layer a new set of features is selected from within the first layer features subset; by analyzing the Pearson correlation coefficients between the selected features and the classes. To demonstrate the superiority of the proposed linear correlation-based FS, several experiments on NSL-KDD datasets are conducted. The experiments shows that the proposed linear correlation-based feature selection method improves the accuracy to 99.1%, while reduces the number of features from 41 to 17 features.

References

1. Tsai, C., Hsu, Y., Lin, C., Lin, W.: Intrusion detection by machine learning: A review. Expert Systems with Applications 36(10), 11994–12000 (2009)
2. Debar, H., Dacier, M., Wespi, A.: Towards a taxonomy of intrusion-detection systems. Computer Networks 31(8), 805–822 (1999)
3. Kuchimanchi, G., Phoha, V., Balagani, K., Gaddam, S.: Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems. In: Proceedings of the Fifth Annual IEEE SMC Information Assurance Workshop, pp. 195–202 (2004)
4. Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., Dai, K.: An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications 39(1), 424–430 (2012)
5. Amiri, F., Yousefi, M., Lucas, C., Shakery, A., Yazdani, N.: Mutual information-based feature selection for intrusion detection systems. Journal of Network and Computer Applications 34(4), 1184–1199 (2011)
6. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering-a filter solution. In: Proceedings of the Second International Conference on Data Mining, pp. 115–122 (2002)
7. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 284–292 (1996)
8. Tsang, C., Kwong, S., Wang, H.: Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition 40(9), 2373–2391 (2007)
9. Elngar, A., Mohamed, D., Ghaleb, F.: A Real-Time Anomaly Network Intrusion Detection System with High Accuracy. Information Sciences Letters International Journal 2(2), 49–56 (2013)
10. Yu, L., Liu, H.: Efficient Feature Selection via Analysis of Relevance and Redundancy. Journal of Machine Learning Research 5(1), 1205–1224 (2004)
11. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 856–863 (2003)
12. Kim, Y., Street, W., Menczer, F.: Feature selection for unsupervised learning via evolutionary search. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 365–369 (2000)

13. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 1(2), 273–324 (1997)
14. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
15. Jin, X., Xu, A., Bie, R., Guo, P.: Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. In: Li, J., Yang, Q., Tan, A.-H. (eds.) BioDM 2006. LNCS (LNBI), vol. 3916, pp. 106–115. Springer, Heidelberg (2006)
16. Ben-Bassat, M.: Pattern recognition and reduction of dimensionality. In: Handbook of Statistics II, vol. 1, North-Holland, Amsterdam (1982)
17. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
18. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
19. Jemili, F., Zaghdoud, M., Ahmed, M.: Intrusion detection based on Hybrid propagation in Bayesian Networks. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, pp. 137–142 (2009)
20. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes in C. The art of scientific computing. Cambridge University Press, Cambridge (1988)
21. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Proceeding of IEEE Symposium on Computational Intelligence in Security and Defense Application, CISDA (2009)
22. KDD'99 dataset, Irvine, CA, USA (July 2010), `http://kdd.ics.uci.edu/databases`
23. Duda, R.O., Hart, P.E., Stork, P.E.: Pattern Classification, 2nd edn. JohnWiley & Sons, USA (2001)

Author Index