IJNS

International Journal of Network Security

ISSN 1816-353X (Print)
ISSN 1816-3548 (Online)

Vol. 17, No. 4 (July 2015)

# INTERNATIONAL JOURNAL OF NETWORK SECURITY

**PUBLISHING OFFICE**

**Min-Shiang Hwang**
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Network Security is published both in traditional paper form (ISSN 1816-353X) and in Internet (ISSN 1816-3548) at http://ijns.jalaxy.com.tw

**PUBLISHER: Candy C. H. Lin**

# International Journal of Network Security

**Vol. 17, No. 4 (July 1, 2015)**

# Secure and Self-healing Control Centers of Critical Infrastructures using Intrusion Tolerance

Maryam Tanha[1], Fazirulhisyam Hashim[2], and Shamala Subramaniam[3]
*(Corresponding author: Maryam Tanha)*

Department of Computer Science, University of Victoria[1]
Victoria, BC, Canada, V8P 5C2
(Email: tanha@uvic.ca)
Department of Computer & Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia[2]
43400 UPM Serdang, Selangor, Malaysia
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia[3]
43400 UPM Serdang, Selangor, Malaysia

*(Received Nov. 14, 2014; revised and accepted Jan. 16 & Mar. 6, 2015)*

## Abstract

Nowadays, critical infrastructures are highly integrated with state-of-the-art information and communication technologies to enhance their efficiency. Due to far-reaching societal and economic impacts caused by failure or malfunction of critical infrastructures, cyber security and self-healing capability are among their salient features. A new security paradigm referred to as intrusion tolerance is envisaged to complement the existing security solutions (i.e., intrusion prevention and detection), as well as to provide availability and self-healing capabilities, particularly for the control centers as the key components of critical infrastructures. However, intrusion tolerance techniques are associated with substantial cost. In this paper, we propose an intrusion tolerant system architecture which incorporates distinctive features, namely dynamic redundancy level, and hybrid and hierarchical rejuvenation mechanism. The acquired results from security analysis of the proposed architecture show improvements compared to two established architectures. Also, analysis of the incurred cost demonstrates the cost-effectiveness of the proposed architecture.

*Keywords: Control center, critical infrastructure, intrusion tolerance, self-healing*

## 1 Introduction

In recent decades, the growing dependence of critical infrastructures on Information and Communication Technology (ICT) and open standards has raised serious concerns about security issues. Critical infrastructures are complex physical and cyber-based systems and assets that lay the foundations for a modern society, and their secure and dependable operation is of utmost importance for national security and economy. Smart grid (as the modern power grid), public health system, and transport system are examples of critical infrastructures. Cyber systems serve as the backbone of critical infrastructures, thus cyber security incidents may not only affect the cyber domain but also potentially impact their dependent physical systems [37]. The cyber-physical dependencies, large-scale operation, heterogeneity and complexity along with sophisticated and novel attacks pose grave and new threats to the mission critical applications in critical infrastructures.

Using open standard software and protocols have opened avenues for attackers to pose dire threats to different sections of critical infrastructures' communication system, particularly SCADA and control systems. Some of the recent high-profile attacks such as Stuxnet worm [16,28] and FLAME [29] have been mainly targeted at control systems of critical infrastructures and crucial organizations. Moreover, the security objectives of critical infrastructures differ from the ICT security goals in their order of significance. Availability, continuity of service and safety are the main security priorities in critical infrastructures.

On top of all the mentioned issues, the widespread and socio-economic impacts of malfunction or failure of critical infrastructures resulting from accidental or malicious events mandate more automatic and robust security solutions [5, 40]. These security approaches can be associated with self-healing capabilities of critical infrastructures. Self-healing responses to malicious acts of sabotage and natural calamities is one the essential features of critical infrastructures. Self-healing is defined as the attribute of a system to be able to recognize abnormal operation (the disturbances may result from security intrusions) and subsequently making proper adjustments to restore to normal conditions [11]. Control centers are considered as the brain of critical infrastructures. They are in charge of data analysis and decision making. For instance, in smart grid as a critical infrastructure [2], based on the assembled data, the control centers make appropriate adjustments to power supply to satisfy demand as well as spot and respond to the defects or failures by sending control commands to field devices. Figure 1 illustrates a control center which supervises other sections in a critical infrastructure. This figure also depicts some of the key components of the control center such as SCADA servers and historian databases.

SCADA systems (as the key components of control centers) play a pivotal role in the proper operation of critical infrastructures, any malfunction or failure of these systems and their underlying software systems may result in widespread and devastating effects on industry, economy and people's daily life.

Therefore, the correct functioning of SCADA systems in exigent security circumstances is of paramount importance. Two of the dire threats to SCADA are Denial of Service (DoS) and unauthorized access/integrity breach [23]. These threats will result in the unreliability of the control signals from the monitoring system as well as the collected data gathered from different sections of critical infrastructures that are used for decision making or other purposes.

In addition, thanks to the time-criticality of the communication and control in some critical infrastructures such as smart grid, a delay of a few seconds (following from an availability attack) may lead to irreparable harm to the national economy and security [18].

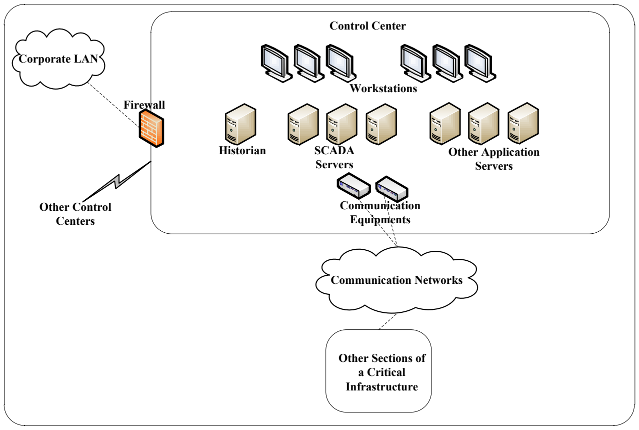


Figure 1: Control center in critical infrastructures

The aforementioned security concerns serve as contributing factors to change our mind set about the level of security that can be achieved through conventional security approaches (i.e., prevention and detection [15]) especially for critical infrastructures.

To satisfy the mentioned security requirements, a promising mechanism called intrusion tolerance has come to existence and it has received considerable attention in recent years [3–5, 7, 20, 22, 26, 27, 31, 38, 40, 47]. Intrusion tolerance is concerned with the fact that it is always probable for a system to be vulnerable to security compromise as well as for some attacks to be launched successfully on a system [40]. In spite of these assumptions, intrusion tolerance mechanisms ensure that the system prolongs its normal activities (or acts in a degraded mode providing only essential services) even when it is under attack or partially compromised. Thus, rather than preventing intrusions from happening in the system, they are tolerated by adopting and triggering appropriate mechanisms such as redundancy, diversity, rejuvenation, and so on.

In this paper, we propose an Intrusion Tolerant System (ITS) architecture to enhance the availability and self-healing capabilities of critical infrastructures while decreasing the associated cost with intrusion tolerance techniques.

The main contributions of our research can be summarized as follows:

- We highlight the importance of intrusion tolerance approach which raises the possibilities for enhancing the security of crucial components in critical infrastructures, particularly control centers and Supervisory Control and Data Acquisition (SCADA) systems.

- An ITS architecture is proposed to enhance the level of security in control centers of critical infrastructures.

- To provide the availability and self-healing capabilities required by critical infrastructures, special focus is placed on redundancy and rejuvenation as two intrusion tolerance techniques. Also, we propose dynamic redundancy level and hybrid and hierarchical recovery algorithms to alleviate the substantial cost associated with these techniques.

The paper is organized as follows. Section 2 provides a detailed analysis of intrusion tolerance as a promising security solution for critical infrastructures. Moreover, the most commonly used intrusion tolerance techniques are presented and a comparison is made between some of existing ITS architectures. In Section 3, a detailed discussion on the proposed intrusion tolerant architecture (its modules and embedded algorithms) for control centers of critical infrastructures is presented. The security and cost analysis of the proposed ITS architecture is provided in Section 4. Finally, Section 5 draws the conclusion. It should be noted that the terms recovery and rejuvenation are used interchangeably throughout this paper.

## 2 Intrusion Tolerance for Critical Infrastructures

Intrusion tolerance is commonly referred to as the third generation of security technologies [14] which provides complementary features to conventional security mechanisms, i.e., prevention and detection. It shows enormous potential to be adopted and deployed in critical infrastructures' control centers in which the correct service and availability is of great importance. The impacts of availability violation in critical infrastructures are substantial and affect the physical world. The possible consequences of service disruption in critical infrastructures range from financial loss to human loss. As an instance in the context of smart grid, a compromised server in control center may result in sending misleading data to the field device. This attack affects the availability (i.e., not allowing unauthorized access and providing correct service). The viable consequences may be equipment damage (if control commands that are sent to the field device lead to overload conditions), blackouts or safety issues (if a line is energized while linemen are in the field servicing the line). To increase the availability of critical infrastructures' control centers the self-healing capabilities are essential. Recovery mechanisms enable the self-healing feature for critical infrastructures, thus in this paper we placed especial focus on rejuvenation mechanisms in order to enhance the availability.

Intrusion tolerance and its paradigms (e.g., replication and recovery) enable secure and normal operation of the control centers of critical infrastructures, even when the system is being attacked or partially compromised. The primary goal of intrusion tolerance is to tolerate malicious events and sustained attacks as well as masking, removing or recovering from intrusions. Thus, intrusion tolerance measures avert security failures and aid to maintain the availability of the system. Moreover, intrusion tolerance places emphasis on the impact of the attack rather than the cause of it [41].

During the last decade, various research have been conducted on intrusion tolerance and multiple intrusion tolerant architectures with specific features and applications have been proposed. The Willow architecture [17], COCA [48], DIT [39], MAFTIA [34], SITAR [44], SCIT [3], Crutial [5], FOREVER [30] and Generic intrusion tolerant architecture for web servers [27] exemplify a number of the proposed ITS architectures. Some of these architectures are application-specific. For instance, the goal of COCA is to provide a secure and fault-tolerant Certification Authority (CA) while Crutial is a distributed firewall-like intrusion tolerant system for critical infrastructures protection such as power grid. But primarily, enhancing the security and availability of distributed services, Commercial Off The Shelf (COTS) servers and critical information systems have called for designing such architectures.

There are several intrusion tolerance techniques that are commonly used in intrusion tolerant systems. Some of the main techniques are as follows:

- Replication: Space redundancy or replication involves physical resource redundancy which is a key building block of many intrusion tolerant systems.

- Diversity: Replication suffers from the underlying problem of fate sharing for replicas [33, 43]. If an attacker discovers and exploits a vulnerability in one replica, it is highly likely that all replicas are susceptible to the same threat. Thus, diversity (usually in its most common form which is operating system diversity [10]) serves as a solution to alleviate this problem.

- Rejuvenation: Rejuvenation involves the restoration of a replica to a pristine state to eliminate the likely effects of intrusions or faults [43]. It can be triggered reactively following from intrusion detection or carried out proactively and periodically.

- Voting: Voting algorithms are employed to reach a consensus on the valid and final output of non-faulty replicated components in an ITS. Using Byzantine Fault Tolerance (BFT) agreement protocols or some criteria such as edit distance (e.g., hamming distance) and hash codes make the comparison feasible. Voting contributes to masking and tolerating intrusions [43].

- Secret Sharing: Secret sharing or threshold scheme is based on the idea of concealing a piece of information by splitting it into several shares and distributing among participants in a manner that specific subsets of the shares are required to rebuild the initial data [1, 6, 25]. This intrusion tolerance technique has been used in ITS architectures, e.g., COCA (a distributed certification authority) and its main purpose is providing confidentiality and integrity. Since in this paper our main goal is to provide availability and self-healing capabilities as the top security priorities for the critical infrastructures, we do not include secret sharing method in our proposed architecture.

- Proxy: Proxies serve as additional layers of defense between replicated servers and clients.

# 3 Proposed ITS Architecture

Typical intrusion tolerant systems have single primary focus. For instance, Scalable Intrusion Tolerant Architecture for Distributed Services (SITAR) is detection triggered, and Self Cleansing Intrusion Tolerance (SCIT) is recovery based. Some ITS architectures (e.g., Crutial) apply a hybrid rejuvenation approach (i.e., both proactive and reactive recovery) that enhances the level of security, but the complexity and cost of redundancy and recovery increases enormously. Based on our feasibility studies, using adaptive redundancy as well as a hybrid and hierarchical rejuvenation approach assists in reducing the incurred cost. Also, the specific requirements of the critical infrastructures' control centers (e.g., self-healing capabilities, delay sensitivity) underscore the need for a new ITS architecture that suits these systems.

By securing the software systems that manage the sub systems of the control centers, we would be able to mitigate the consequences of cyber security incidents that

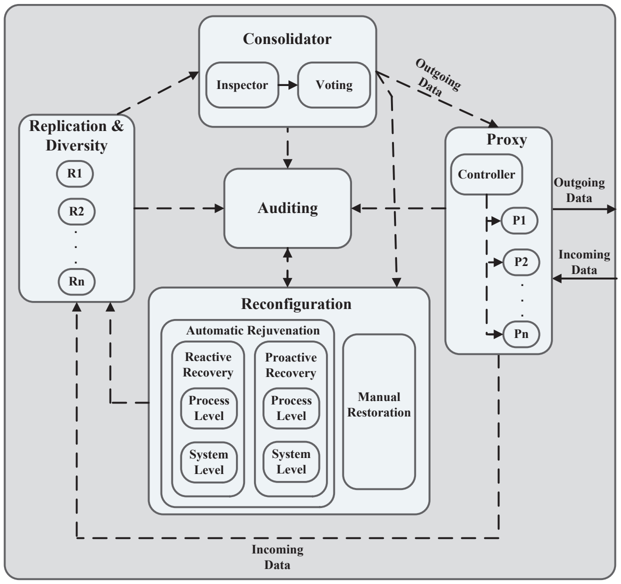


Figure 2: The proposed ITS architecture

affect the physical domain (e.g., blackouts in smart grid) and subsequently, enhancing the cyber-physical security of critical infrastructures' control centers. Our proposed ITS encompasses a rich blend of a wide spectrum of different intrusion tolerance techniques. As illustrated in Figure 2, the proposed system comprises five modules, namely replication & diversity module, auditing module, consolidator module, reconfiguration module and proxy module. The role and working principles of the aforementioned modules are elucidated in the following sections. In general, our proposed system is a security architecture that can be hosted by a dedicated server (including auditing module, consolidator module, reconfiguration module and proxy module) that manages a number of physical/virtual replicas (in replication & diversity module). These replicas run one or more critical applications in control centers as well as agents of the proposed ITS.

It should be noted that in the proposed ITS architecture the emphasis is placed on offering algorithms for automatic and hierarchical rejuvenation as well as managing replication and rejuvenation mechanisms cost-effectively. This is due to the importance of availability and self-healing capabilities for the critical infrastructure along with addressing the issue of substantial cost incurred by intrusion tolerance techniques. In essence, recovery-oriented computing is a vital aspect of a self-healing system [11] such as critical infrastructure. Intrusion tolerance techniques such as redundancy and rejuvenation contribute towards provisioning availability and self-healing characteristics. Moreover, to avoid the proposed ITS from being compromised by the intruders, it is assumed that all the components' tasks and their communications are performed in a trusted platform. Proxy module also helps to enhance the security of the ITS.

## 3.1 Replication & Diversity Module

Replication in ITSs is usually integrated with Byzantine Fault Tolerance (BFT) algorithms in which the number of replicas is required to be $3f+1$ to tolerate $f$ faulty replicas. As a result, a fault/intrusion tolerant distributed system is obtained which is enabled to tolerate $f$ Byzantine (i.e., arbitrary) faults. The aforementioned arbitrary faults model accidental faults or malicious attacks and intrusions. Specifically, the key idea of BFT algorithms is to enable a system to automatically continue correct operation despite the fact that some of its components show arbitrary, probably malicious behavior. BFT algorithms have already been adopted to design intrusion tolerant services such as network file systems, cooperative backup, large scale storage and certification authorities [42].

The replication & diversity module consists of a number of replicas for a critical entity (usually a physical server running crucial applications such as Master Terminal Unit (MTU) or historian databases as shown in Figure 1) in the control centers of the critical infrastructures. In addition, with regard to different levels of security needed in different points of the critical infrastructures, this module can be modified accordingly. The replicas can be physically distributed (i.e., in different machines) for application such as automatic grid separation in emergency states. As another example in the context of smart grid, the replication module can be utilized in substations with replicas as virtual machines running in the same host. Although the system does not tolerate physical faults, it may provide adequate protection for substations in smart grid. Similar approach has been used in [5].

In this module, the number of replicas is assumed to be $2f+1+k_{max}$ (to tolerate $f$ faulty/compromised replicas provided that there are trusted components) and the value of $f$ and $k_{max}$ ($f, k_{max} \geq 1$) are indicated in the deployment time. A similar approach also used to design a distributed firewall-like protection device named Crutial Information Switch (CIS) in [5]. $k_{max}$ denotes the maximum possible number of concurrent recoveries. The reason why the value of $k_{max}$ is added to the number of replicas will be discussed in the reconfiguration module section.

As mentioned before, diversity decreases the possibility of being vulnerable to the same intrusion for different replicas. In the proposed ITS, all replicas have operating system diversity to decrease the probability of sharing the same vulnerabilities. Operating system is considered a vital element of each replica on account of hosting the SCADA system and other critical components in control centers. Subsequently, any misconfiguration or vulnerability in it may bring down the SCADA system and causes the adversaries achieve breakthroughs [23]. Hence, operating system diversity proves an appropriate approach for applying diversity to replicas. However, the number of existing and tailored operating systems are limited, thus the diversity level is confined to this number. To have an effective system using both redundancy and diversity, the redundancy level and diversity level are expected be equal. More specifically, if the total number of replicas is assumed to be $2f+1+k_{max}$, the ideal degree of diversity should be the same. By using a different operating system in each replica, it is less probable that the replicas suffer from similar vulnerabilities. If the diversity degree is less than redundancy degree, at least two replicas will have the same operating system and consequently will experience the same fate in the event of intrusions. In contrast, if we assume the diversity degree more than redundancy degree, part of the diversity level is useless since it is not applied to any replica.

## 3.2 Consolidator Module

In this module, the outputs are inspected and then consolidated into one. More specifically, this module aims at examining the responses/outputs of the replicas to identify possible infected/compromised ones. It is composed of the following sub modules.

- Inspector: Acceptance testing [45] as an intrusion tolerance technique is entailed in the inspector module. It involves application-specific checks with regard to the security policy to ensure the sanity of outgoing data (e.g, control commands that are sent to substations in critical infrastructure) from the replicas. Any symptom of security compromise detected by it will trigger the reactive recovery sub module in the reconfiguration module. In contrast with SITAR, acceptance testing is only performed on the outgoing data in our proposed ITS architecture. This would result in decreasing the delay imposed by the proposed ITS for processing the incoming data in critical systems such as smart grid control systems that are delay sensitive. The incoming data must satisfy the time requirements otherwise it is not useful. Also, some preliminary check on the incoming data can be performed by proxies. In contrast, due to the crucial importance of the outgoing data which are usually the control commands in critical systems their sanity and correctness should be tested before letting them leave the system.

- Voting: This sub module is intended for masking the impacts of intrusions as well as ensuring the integrity of replicas outputs. Based on a voting algorithm, it seeks for the correct output by comparing the redundant outputs from the active replicas that passed the inspector. In this way, it will arrive at a consensus on the final desired output to be passed to the proxy module. This output can be a command or information from the control centers destined for a device or component in critical infrastructures.

## 3.3 Reconfiguration Module

Reconfiguration module consists of two sub modules namely, automatic rejuvenation and manual restoration. When the proposed ITS is able to mask an intrusion, it uses the automatic rejuvenation sub module, otherwise it takes advantage of restoration which involves human intervention. Manual restoration happens when for instance the system is targeted by DoS attacks and only capable of provisioning the essential services. The sub modules descriptions are provided in the following sections.

**Automatic rejuvenation:** Automatic rejuvenation mostly addresses the required self-healing capabilities of critical infrastructures. In the event of detecting an abnormal behavior of a replica or periodically, it triggers a recovery for the respective replica. Also, automatic rejuvenation enables the concurrent rejuvenation of at most $k$ ($1 \leq k \leq k_{max}$) replicas out of $2f+1+k_{max}$ (total number of replicas). The assumption for the total number of replicas eliminates the impact of compromised replicas (at most $f$) and recovery on the availability of the system. It should be noted that $k$ has a fixed value in Crutial (usually $k=1$) whereas the value of $k$ is dynamic in our proposed ITS architecture.

In this module, a hybrid rejuvenation approach, i.e., proactive and reactive recovery, has been used to address the shortcomings of the two aforementioned rejuvenation approaches. Reactive recovery mainly relies on the underlying intrusion detection methods and subsequently is subjected to the same drawbacks such as inability to detect unknown attacks and false positives. In contrast, proactive recovery can compensate for dormant or undetected intrusions. By assuming an asynchronous distributed system model and proactive recovery and it is not possible to guarantee that recoveries are performed within known time bounds. Thus, we have used a hybrid distributed system model that uses some trusted components to ensure that replicas are always rejuvenated in accordance to predefined time bounds.

A hybrid rejuvenation mechanism will enhance the performance of the system through decreasing the possible duration of time a compromised replica may disrupt the normal operation of the system [31].

To come up with a cost-effective and hybrid rejuvenation mechanism, we were inspired by a hierarchical reactive recovery method that has been proposed recently in [14]. This model eliminates the need for complete recovery when the system is partly compromised. The merits of this model can be considered as reduced total recovery time, improved flexibility and dependability.

In the proposed ITS architecture, reactive recovery can be triggered externally and at the system level by the consolidator module or internally (within a replica) in a hierarchical fashion (including process level recovery and system level recovery). Proactive recovery is performed periodically by choosing an active replica based on smallest rejuvenation time stamp in a hierarchical manner. It is triggered by the proactive recovery sub module of the reconfiguration module. The details of the proposed rejuvenation algorithm are provided in Section 3.6.

**Manual restoration:** This sub module is triggered when the intrusion (whether detected or not) is nonmaskable (e.g., more than $f$ replicas have been compromised). This may cause the system to be in graceful degradation mode, stopped functioning mode or complete failure mode all of which require human intervention and corrective measures to return to the normal working state.

### 3.4 Auditing Module

This module maintains audit logs for all modules. The logs would be useful for security administrator to monitor and analyze the operation of the system.

### 3.5 Proxy Module

The Proxy module is placed on the boundary of the ITS architecture where the data comes in or goes out. The proxy module shields the internal structure of the ITS from attackers as well as acting as a load balancer.

The incoming data go through the proxy module as the first layer of defense. This data is then forwarded to the replication & diversity module to be dealt with. Moreover, the control commands from the SCADA system (outgoing data in Figure 2) pass the proxy to reach the devices or other components of critical infrastructures.

Proxy module is composed of several proxies located in different virtual machines that have diversity in their operating systems and are managed by a controller. Proxies can have three modes, namely online, offline, and cleansing. The number of online proxies can be one or more based on the decision of the controller. Depending on a defined exposure time for proxies and a round-robin algorithm, the controller deals with the rotation and changing turn between proxies [3]. When the exposure time requirement for a proxy is met, it will go through the rejuvenation process (or cleansing process) and will be in cleansing mode. Then, its mode will be altered to offline mode and it will be ready to be chosen by the controller to go online.

### 3.6 Cooperative Operation of Replication and Reconfiguration Modules

In the proposed ITS architecture, we mainly focus on two intrusion tolerance techniques, namely redundancy and rejuvenation. The cooperative operation of replication and rejuvenation module provides availability and self-healing features in a cost-effective manner. This would assist in enhancing the level of security while reducing the cost.

**Hybrid and hierarchical rejuvenation algorithm:** While in [14] the hierarchical recovery is performed at three levels and only applies to reactive recovery, our proposed algorithm employs a hierarchical recovery strategy at two levels (i.e., process level and system level) for both proactive and reactive rejuvenations. It should be noted that a process is defined as an instance of a computer program. A process can be a system process such as a background process for logging and monitoring or an application process such as Internet Explorer or any application that is running on a SCADA server or other application servers in a control center in critical infrastructure. Algorithm 1 shows the proposed hybrid and hierarchical rejuvenation mechanism which is provided by the cooperation of replication & diversity and reconfiguration modules. Process manager is a module executed in each active replica to handle the process level recovery. At the deployment time, critical processes in the replicas are identified. With regard to this, there are two sets of processes, namely active set (includes running processes) and standby set. To differentiate between reactive process level recovery and proactive process level recovery, the process manager includes two components, namely PLRR (Process Level Reactive Recovery) and PLPR (Process Level Proactive Recovery).

PLRR (Line 1 in Algorithm 1) acts as a type of host-based IDS which features self-healing capabilities. Based on a timeout period, it examines the pool of active processes. In the event of finding any suspected process, PLRR will obtain the relevant checkpoint, kills the process and activates its peer from the standby set (if there is any) otherwise the system level reactive recovery (SLRRTriggered denotes a system level reactive recovery) may be performed.

PLPR (Line 15 in Algorithm 1) deals with process level proactive recovery. After a timeout period which is set in the proactive recovery sub module of the reconfiguration module, a proactive rejuvenation signal is sent to the replica with the least rejuvenation time stamp. In this case, if there are standby processes available for all the critical active processes, the PLPR will replace each active process with its peer from the standby set in a similar way to PLRR. However, if the aforementioned condition is not satisfied, the proactive recovery (SLPRTriggered indicates a system level proactive recovery) may be carried out at system level for the respective replica.

The process level recovery is time-saving compared to system level recovery as well as it is more secure since it mainly involves internal information and communication exchange in a machine. Moreover, it does not require the replica to go offline for performing the recovery and causes less overhead on the replica.

**Dynamic redundancy level algorithm:** The total number of replicas represents the redundancy or diversity level (we assume that they have the same value). The redundancy level is a linear and increasing function of $f$ and $k$. However, increasing $f$ will have more impact on the

**Algorithm 1** Hierarchical recovery

```
Begin
if $PLRR - timeout$ then
    $Detection\&Polling()$;
    for all the suspected processes (j) in replica i do
        if $Process[j].StandbyAvailable()$ then
            $Process[j].ObtainCheckpoint(Suspect)$;
            $Process[j].Kill(Suspect)$;
            $Process[j].ActivateStandby()$;
            $Reset\ respective\ recovery\ timer$
        else
            $Replica[i].SLRRTriggered = True$;
            $Exit\,the\,for\,loop$
        end if
    end for
end if
if $PLPR - timeout$ then
    $Polling()$;
    for all the processes (j) in replica i do
        if $Process[j].StandbyAvailable()$ then
            $Process[j].ObtainCheckpoint(Suspect)$;
            $Process[j].Kill(Suspect)$;
            $Process[j].ActivateStandby()$;
            $Reset\ respective\ recovery\ timer$
        else
            $Replica[i].SLPRTriggered = True$;
            $Exit\,the\,for\,loop$
        end if
    end for
end if
End
```

increase of the number of replicas compared to $k$. This means that increasing $f$ incurs more resource cost. Having less impact on the resource cost, increasing the value of $k$ would lead to increase in the cost of recovery which may be more preferable to the resource cost increase following from increasing the value of $f$.

Having a dynamic redundancy level is regarded as an effective way to cut the resource cost and recovery cost to a certain extent. An adaptive redundancy level algorithm has been offered in [27] based on attack and alert severity. It provides a triplex regime for critical applications, i.e., three web servers out of $N$ web servers process each client request. This regime can be dynamically increased provided that the attack rate increased. After a specific period of time with no compromise detection, the regime will be decreased. Three algorithms for dynamic redundancy design have been presented in [24] for NAN (Neighbourhood Area Network) gateways in critical infrastructure which considered availability threshold and cost minimization. Our proposed dynamic redundancy algorithm is integrated with recovery mechanism and handled by the automatic rejuvenation module. Considering the possible maximum redundancy/diversity level ($MaxRL = MaxDL = 2f + 1 + k_{max}$) as an upper bound for the total number of replicas, the offered adaptive redundancy level algorithm involves increasing the value of $k$ (Line 7 in Algorithm 2) based on the number of system level recovery signals sent to the reconfiguration module. If the number of ongoing rejuvenations is equal to $k$ and $RL < MaxRL$ ($RL$ denotes redundancy level and it is equal to $2f + 1 + k$ which is the number of active or under recovery replicas), then a new incoming system level rejuvenation signal would result in increasing the number of $k$ by one, otherwise (if $k = k_{max}, i.e, RL = MaxRL$) the recovery signal is ignored by the automatic rejuvenation module. After a time-out period, if the number of ongoing rejuvenations is less than $k$, this value is reduced by 1. We can also conclude that the number of standby replicas is calculated by subtracting the value of ($RL$) from the value of ($MaxRL$) in each moment. Algorithm 2 illustrates the adaptive redundancy algorithm.

**Algorithm 2** Dynamic redundancy level

```
 1: Begin
 2: $k = 1$;
 3: $RL = 4$;
 4: $MaxRL = 6$;
 5: while true do
 6:     if $SLRRTriggered \parallel SLPRTriggered$ then
 7:         if $OngoingRejs = k\ \&\&\ RL < MaxRL$ then
 8:             $k + +$;
 9:             $RL + +$;
10:             $OngoingRejs + +$;
11:             $Reset\ respective\ recovery\ timer$
12:         else
13:             if $OngoingRejs < k$ then
14:                 $OngoingRejs + +$;
15:             end if
16:         end if
17:     end if
18:     if $RL - timeout$ then
19:         if $OngoingRejs < k$ then
20:             $k - -$;
21:             $RL - -$;
22:         end if
23:     end if
24: end while
25: End
```

### 3.7 A Case Study: Trojan Horse Attack on Smart Grid Control Centers

For the benefits of readers we describe the working principle of the proposed ITS architecture by an attack scenario in the context of smart grid. Suppose a possible intrusion scenario in which an attacker (an outsider or a malicious insider) has bypassed prevention or even detection mechanisms and has gained access to the SCADA system in smart grid (this attack can be considered as privilege escalation). Subsequently, he/she tries to infect one or more replicas of a critical component. Figure 3

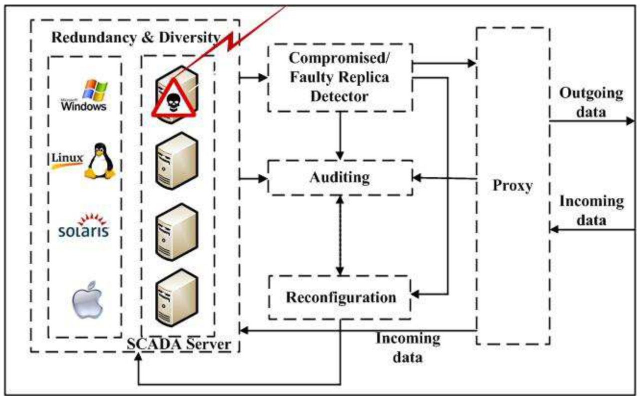


Figure 3: Case study: Trojan horse

depicts a SCADA server that has been compromised by a Trojan horse. It is less probable that more than one replica be infected by this attack since there is diversity in the operating systems of the replicas (all of them are not vulnerable to the same type of attack). We can consider more than one type of attack (not only Trojan horse) on the system but due to the diverse replicas it is highly unlikely that more than one replica affected by the same type of attack. As long as the current redundancy level is less than or equal to the maximum allowed by the ITS, there can be faulty replicas that have been intruded even by more than one type of attack (whether unknown attack or known attack). It is possible that the adversary causes the replica become malfunctioned by running a Trojan and changing some system files which may result in sending inappropriate control commands (in case of automatic operation). However, the command must first pass the consolidator. It is highly probable that the compromised replica(s) being recognized (due to the fact that the replicas have different operating systems, all of them may not be infected by the same attack targeted at a special type of vulnerability, and thus the generated responses would be different) by the inspector (using detection capabilities) and the infected replicas would undergo reactive recovery. Another possibility is that the infected outputs may be masked by the voting module. In addition, process manager running in each replica may detect the infection and trigger the process level rejuvenation. Even if the intrusion tolerance mechanisms fail to detect the intrusion, it is possible that the attack's impact is masked through proactive recovery (whether at process level or system level). During the performance of attack masking measures by the proposed ITS, if the number of required concurrent rejuvenations becomes more than current $RL$ ($RL < MaxRL$), the redundancy level will be increased adaptively (at most up to $MaxRL$).

### 3.8 Comparison of Existing ITS Architectures and the Proposed ITS Architecture

The intrusion tolerance techniques can be utilized to analyze and compare different intrusion tolerant architectures. Some representatives of existing ITS architectures have been compared by conducting qualitative analyses in [22,26]. Table 1 depicts such analysis with emphasis on the paradigms of intrusion tolerance employed in several ITSs. The spectrum of architectures have distinct features. In this paper, we conduct a comparative analysis to enable a clear reflection of their respective attributes. Moreover, our provided comparison encompasses a higher volume of ITSs.

As it can be seen in Table 1, replication and diversity are the techniques adopted by almost all of the ITSs. The adaptive redundancy that has been employed in the ITS for web servers and our proposed architecture contributes towards reducing the redundancy cost. Although design diversity (e.g., using different operating systems) is the dominant type of diversity used by the ITSs, FOREVER and Crutial can employ time diversity (i.e., rejuvenation introduces diversity). Some ITSs such as FOREVER, Willow and Crutial apply a hybrid recovery method whereas others like SCIT only use proactive recovery. To the best of our knowledge, none of the existing ITSs utilizes the hierarchical recovery strategy introduced in [14]. However, as mentioned earlier, our proposed ITS architecture uses a hybrid and hierarchical recovery approach which decreases the recovery cost. One of the indirection techniques that is widely preferred is the use of proxies as the mediator between the COTS servers and the outside network. Intrusion detection methods whether anomaly-based or signature-based are very common among the ITSs. Byzantine agreement algorithms and secret sharing are the other intrusion tolerant mechanisms that have been implemented in some of the architectures. Among the ITSs shown in Table 1, SCIT and SITAR have drawn more attention in published intrusion tolerance research and investigated with regard to their performance [9,19,21,38]. In addition, interested readers can find useful hints about implementation of the proposed architecture in existing architectures as stated in [3, 5, 44, 46]. For instance, the SCIT has used Virtual Box as the virtual machine monitor and setup four Ubuntu 9.04 server edition on each machine. This approach may be applied to the replicated servers in our proposed architecture.

## 4 Performance Evaluation

This section is divided into two parts namely, security analysis and cost analysis. Security analysis using a semi Markov model shows the effectiveness of the hybrid rejuvenation approach to enhance the security whereas cost analysis demonstrates the cost-effectiveness of the proposed algorithms.

### 4.1 Security Analysis

Security quantification of the proposed ITS architecture is needed for assessing the outcome of the desired performance measures as well as performance comparison with other architectures. To achieve this goal, a state-space model is developed that incorporates an attacker's behavior along with the system's response to an attack or intrusion [9, 12, 19, 21, 38]. State transition diagrams assist in the evaluation of the transitions impacted by the inter-domain dependencies in the cyber-physical systems. They describe how the attacker's actions cause transitions to failure states [32]. The main advantage of state transition models is the ability to provide a fine-granular system description which includes the dynamic behavior of system [13]. Moreover, these models are tailored to model immense and complex systems such as the critical infrastructures. In this paper we utilize Semi-Markov Process (SMP) which is a generalization of both continuous and discrete time Markov chains which allows arbitrary state holding time distribution functions, probably relying on both the current state and on the state to be visited afterwards [8]. We place focus on the evaluation of the sensitivity of the steady-state availability and Mean Time To Security Failure (MTTSF) as performance measures to variations in model parameters (i.e., $p_{\text{I}}$ and $h_{\text{G}}$). The same approach has been used in [19,21,38]. A comparison between the proposed ITS and two of the existing ITSs, namely SITAR and SCIT has been made using these parameters. The aforementioned architectures have been chosen for comparison with our proposed ITS firstly, due to their main focus on improving availability which is the top security priority in critical infrastructures and secondly due to their popularity compared to other existing ITS architectures. The analytical evaluation has been carried out using MATLAB simulator.

#### 4.1.1 System Model

The derived state transition diagram for the proposed ITS is shown in Figure 4. Considering both reactive and proactive intrusion tolerance measures (particularly in terms of the hybrid rejuvenation mechanism as the case in our proposed ITS and also in Crutial) differentiates this state transition diagram from its peer in [19]. Thus, it can serve as a generic model for analyzing the behavior of various ITS architectures.This state transition diagram incorporates different security related states of the ITS and their respective interrelationships. Table 2 presents these security states and their corresponding descriptions.

The system changes from one state to the other during its functional lifespan following from normal usage, abuse, maintenance and corrective measures, failures, and so on. Therefore, the behavior of the system is portrayed as the transitions between the states and each transition corresponds to a specific event. Since the interval between the transition from one state to the other (i.e., state holding time or inter event time) is inclined to be random, its underlying process is defined as a stochastic process [13]. In our system, this process is associated with arbitrary probability distributions, thus, it can be modeled using an SMP.

Table 2: Different states of the system and their respective descriptions

| State | Description |
|---|---|
| G | Good |
| V | Vulnerable |
| I | Intruded |
| DMC | Detected Masked Compromised |
| UMC | Undetected Masked Compromised |
| UNC | Undetected Not masked Compromised |
| DNC | Detected Not masked Compromised |
| GD | Graceful Degradation |
| FS | Fail-secure |
| F | Failed |

An SMP can be studied by finding the embedded discrete time Markov chain that requires two sets of parameters [12, 19]:

1) Mean sojourn time (i.e., state holding time) for each state;

2) The transition probabilities between different states.

With respect to Figure 4, the Discrete Time Semi Markov Model (DTSMM) possesses a discrete state space $X_s$= {G, V, I, UMC, DMC, DNC, UNC, FS, F} for which $h_i$ indicates the mean sojourn time in state $i \in X_s$ and $p_{ij}$ represents the transition probabilities between states $i$ and $j$ $(i, j \in X_s)$.

#### 4.1.2 Availability Formulation and Analysis

We analyze the sensitivity of the availability with respect to two parameters, including the probability of intrusion ($p_{\text{I}}$) and the mean time to resist becoming vulnerable to intrusions ($h_{\text{G}}$) [35, 36]. The steady-state availability $A$ is defined as the probability that the system is in one of normal functioning states. One approach to determine the availability is to pinpoint what the unavailable states (i.e., states FS, F and UNC) are. Thus, the steady-state availability $A$ can be formulated as,

$$A = 1 - (\pi_{\text{UNC}} + \pi_{\text{FS}} + \pi_{\text{F}}) \quad (1)$$

where $\pi_i$, $i \in \{UNC, FS, F\}$ denotes the steady-state probability of being in state $i$ for the SMP, that can be computed as,

$$\pi_i = \frac{\nu_i h_i}{\sum \nu_j h_j}, \quad i, j \in X_s \quad (2)$$

where $h_i$ indicates the mean state holding time in state $i$ and $\nu_i$ denotes the embedded Discrete Time Markov chain (DTMC) steady-state probability in state $i$. We can derive $\nu_i$s from the following two equations,

$$\begin{aligned} \nu &= \nu \cdot P && (3) \\ \sum_i \nu_i &= 1, \quad i \in X_s && (4) \end{aligned}$$

Table 1: Comparative analysis of intrusion tolerant architectures (Y: Yes, N: No, O: Optional)

| | COCA | DIT | Willow | SITAR | SCIT | MAFTIA | Crutial | FOREVER | ITS for web servers | Proposed architecture |
|---|---|---|---|---|---|---|---|---|---|---|
| Replication | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Diversity | N | Y | Y | Y | O | Y | Y | Y | Y | Y |
| Proactive Recovery | Y | Y | Y | N | Y | N | Y | Y | Y | Y |
| Reactive Recovery | N | Y | Y | Y | N | Y | Y | Y | Y | Y |
| Hierarchical Recovery | N | N | N | N | N | N | N | N | N | Y |
| Voting/BFT Agreement | Y | Y | N | Y | N | Y | Y | Y | N | Y |
| Proxy | N | Y | N | Y | N | N | N | N | Y | Y |
| Intrusion Detection Capabilities | Y | Y | Y | Y | N | Y | Y | Y | Y | Y |
| Secret Sharing | Y | N | N | N | N | Y | N | N | N | N |

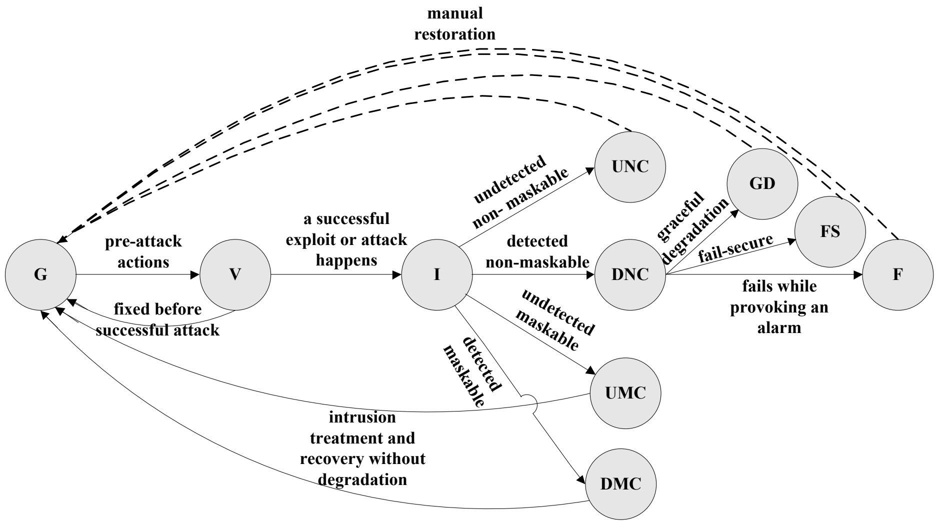


Figure 4: State transition diagram for the proposed ITS

where the $P$ is the transition probability matrix of the corresponding DTMC for the proposed ITS,

$$P = \begin{array}{c|cccccccccc} & G & V & I & DMC & UNC & UMC & DNC & FS & GD & F \\ \hline G & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ V & 1-p_{\mathrm{I}} & 0 & p_{\mathrm{I}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & p_{\mathrm{DM}} & p_{\mathrm{UN}} & p_{\mathrm{UM}} & p_{\mathrm{DN}} & 0 & 0 & 0 \\ DMC & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ UNC & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ UMC & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ DNC & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{\mathrm{FS}} & p_{\mathrm{GD}} & p_{\mathrm{F}} \\ FS & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ GD & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ F & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

In this paper, the mean state holding times $h_i$ for all the states of DTMC have been assumed to have the same values as [19] except for the state UMC which is a new state (corresponding to proactive recovery) for our proposed ITS.

Finally, by using Equations (1)-(4), the steady-state availability ($A_{\mathrm{P}}$) of our proposed ITS is computed as,

$$A_{\mathrm{p}} = 1 - \frac{h_{\mathrm{UNC}}p_{\mathrm{I}}p_{\mathrm{UN}} + h_{\mathrm{F}}p_{\mathrm{I}}p_{\mathrm{DN}}p_{\mathrm{F}} + h_{\mathrm{FS}}p_{\mathrm{I}}p_{\mathrm{DN}}p_{\mathrm{FS}}}{h_{\mathrm{G}} + h_{\mathrm{V}} + p_{\mathrm{I}}(h_{\mathrm{I}} + h_{\mathrm{DMC}}p_{\mathrm{DM}} + h_{\mathrm{UNC}}p_{\mathrm{UN}} + h_{\mathrm{UMC}}p_{\mathrm{UM}} + h_{\mathrm{DNC}}p_{\mathrm{DN}} + h_{\mathrm{GD}}p_{\mathrm{DN}}p_{\mathrm{GD}} + h_{\mathrm{FS}}p_{\mathrm{DN}}p_{\mathrm{FS}} + h_{\mathrm{F}}p_{\mathrm{DN}}p_{\mathrm{F}})} \quad (5)$$

In a similar manner, the steady-state availability for SITAR ($A_{\mathrm{SITAR}}$) and SCIT ($A_{\mathrm{SCIT}}$) are derived as,

$$A_{\mathrm{SITAR}} = 1 - \frac{h_{\mathrm{UNC}}p_{\mathrm{I}}p_{\mathrm{UN}+h_F}p_{\mathrm{I}}p_{\mathrm{DN}}p_{\mathrm{F}} + h_{\mathrm{FS}}p_{\mathrm{I}}p_{\mathrm{DN}}p_{\mathrm{FS}}}{h_{\mathrm{G}} + h_{\mathrm{V}} + p_{\mathrm{I}}(h_{\mathrm{I}} + h_{\mathrm{DMC}}p_{\mathrm{DM}} + h_{\mathrm{UNC}}p_{\mathrm{UN}} + h_{\mathrm{DNC}}p_{\mathrm{DN}} + h_{\mathrm{GD}}p_{\mathrm{DN}}p_{\mathrm{GD}} + h_{\mathrm{FS}}p_{\mathrm{DN}}p_{\mathrm{FS}} + h_{\mathrm{F}}p_{\mathrm{DN}}p_{\mathrm{F}})} \quad (6)$$

$$A_{\mathrm{SCIT}} = 1 - \frac{h_{\mathrm{F}}p_{\mathrm{I}}p_{\mathrm{F}}}{h_{\mathrm{G}} + h_{\mathrm{V}} + p_{\mathrm{I}}(h_{\mathrm{I}} + h_{\mathrm{UMC}}p_{\mathrm{UM}} + h_{\mathrm{F}}p_{\mathrm{F}})} \quad (7)$$

It should be pointed out that some of the transition probabilities may have different values or even may not be applicable for all three ITSs. This follows from the fact that the three ITSs do not possess the same state space (DTSMM's state space for SITAR does not include state UMC whereas SCIT does not contain the states DMC, DNC, UNC, GD and FS).

In Figure 5, the availability for SCIT falls sharply when the probability of intrusion increases compared to the other two ITSs. This is due to the fact that SCIT lacks detection capabilities and only uses periodic rejuvenation. SCIT may alleviate the impacts of attacks on the system, but it does not identify the type of attack (if it is a known attack) to deal with it more appropriately (e.g., triggering recovery when an attack is detected). However, its advantage is dealing with unknown attacks. Considering Figure 5, availability performance of the proposed ITS shows 0.6% and 36% improvement compared to SITAR and SCIT respectively. Figure 6 shows the positive impact of increasing the time that the system is in the good state on the availability (i.e., the availability increases as the $h_{\text{G}}$ rises). For larger values of $h_{\text{G}}$, there is a slight difference in availability performance of the three ITS. In this figure, availability performance of the proposed ITS presents 0.3% and 9% improvement compared to SITAR and SCIT respectively. This is mostly due the use of the hybrid and hierarchical recovery approach in the proposed ITS. While proactive recovery (reflected in state UMC in Figure 4) deals with dormant faults in the system or unknown attacks against the system, reactive recovery (reflected in state DMC in Figure 4) eliminates the effects of known attacks on the system.

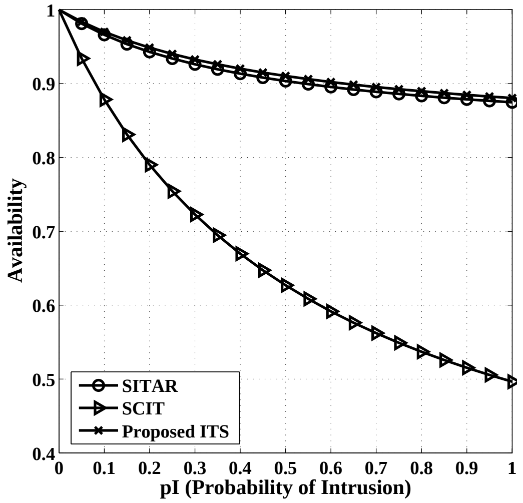


Figure 5: Impact of probability of intrusion on availability

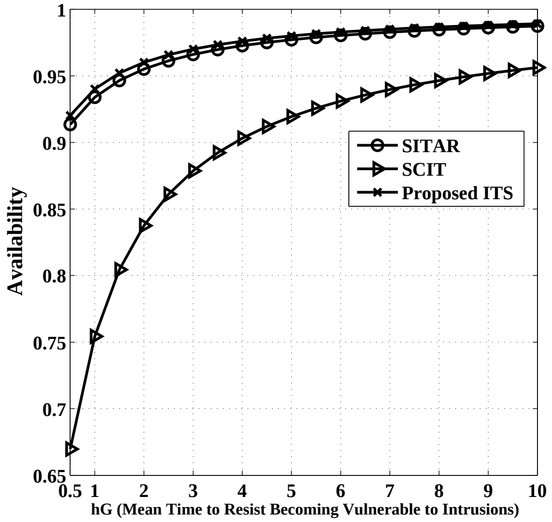


Figure 6: Impact of state holding time in state G on availability

#### 4.1.3 MTTSF Formulation and Analysis

MTTSF is defined as the mean elapsed time for the system to reach one of the security-compromised states (also called absorbing states), provided that the system begins in state G [19]. In the context of critical infrastructure, it can demonstrate the resiliency and robustness of the proposed ITS for control centers' critical components such as SCADA and application servers. A secure and robust ITS is expected to have a high MTTSF when facing intrusions. Using a similar approach to availability analysis, we analyze the MTTSF with regard to $p_{\text{I}}$ and $h_{\text{G}}$ parameters. We take advantage of an SMP with absorbing and transient states. In the state transition diagram shown in Figure 4, the set of states $X_a$ = {UNC, GD, FS, F} are considered as the absorbing states (i.e., the probability of moving out of these states is zero). These states indicate the security compromised states. The rest of the states are called transient states and denoted by $X_t$ = {G, V, I, UMC, DMC, DNC}. The transition probability Matrix $M$ exhibits the transition probabilities between the transient states (i.e., $Q$) and the states originating from transient states to absorbing states (i.e., $C$) in an organized form.

$$M = \begin{pmatrix} Q & | & C \\ -- & | & -- \\ 0 & | & I \end{pmatrix}$$

Matrixes $Q$ and $C$ are as follows:

$$Q = \begin{array}{c} \\ G \\ V \\ I \\ DMC \\ UMC \\ DNC \end{array} \begin{array}{cccccc} G & V & I & DMC & UMC & DNC \end{array} \hspace{-1em} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1-p_{\text{I}} & 0 & p_{\text{I}} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{\text{DM}} & p_{\text{UM}} & p_{\text{DN}} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$Q = \left(\begin{array}{c|cccccc} & G & V & I & DMC & UMC & DNC \\ \hline G & 0 & 1 & 0 & 0 & 0 & 0 \\ V & 1-p_{\text{I}} & 0 & p_{\text{I}} & 0 & 0 & 0 \\ I & 0 & 0 & 0 & p_{\text{DM}} & p_{\text{UM}} & p_{\text{DN}} \\ DMC & 1 & 0 & 0 & 0 & 0 & 0 \\ UMC & 1 & 0 & 0 & 0 & 0 & 0 \\ DNC & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$C = \left(\begin{array}{c|cccc} & UNC & FS & GD & F \\ \hline G & 0 & 0 & 0 & 0 \\ V & 0 & 0 & 0 & 0 \\ I & p_{\text{UN}} & 0 & 0 & 0 \\ DMC & 0 & 0 & 0 & 0 \\ UMC & 0 & 0 & 0 & 0 \\ DNC & 0 & p_{\text{FS}} & p_{\text{GD}} & p_{\text{F}} \end{array}\right)$$

Then we can compute the MTTSF by the following formula [19],

$$MTTSF = \sum_{i \in X_t} V_i h_i \qquad (8)$$

where $V_i$ indicates the average number of times the transient state $i$ has been visited before the DTMC arrives at one of the absorbing states and $h_i$ indicates the mean state holding time in state $i$.

Let $q_i$ be the probability of start in state $i$ (here, it is assumed that the DTMC starts in state G) and $q_{ji}$ be the transition probability from the transient state $j$ to the transient state $i$. So, the $V_i$s can be computed through solving the system of equations,

$$V_i = q_i + \sum_j V_j q_{ji}, \quad i, j \in X_t \qquad (9)$$

Finally, we use Equation (8) to calculate the MTTSF for the proposed ITS as,

$$M_{\mathrm{P}} = \frac{h_{\mathrm{G}}p_{\mathrm{I}}^{-1} + h_{\mathrm{V}}p_{\mathrm{I}}^{-1} + h_{\mathrm{I}} + h_{\mathrm{DMC}}p_{\mathrm{DM}} + h_{\mathrm{UMC}}p_{\mathrm{UM}} + h_{\mathrm{DNC}}p_{\mathrm{DN}}}{1 - p_{\mathrm{DM}} - p_{\mathrm{UM}}} \tag{10}$$

Using the same approach, we derive the expression for SITAR [19] and SCIT as follows,

$$\begin{aligned} M_{\mathrm{SITAR}} &= \frac{h_{\mathrm{G}}p_{\mathrm{I}}^{-1} + h_{\mathrm{V}}p_{\mathrm{I}}^{-1} + h_{\mathrm{I}} + h_{\mathrm{DMC}}p_{\mathrm{DM}} + h_{\mathrm{DNC}}p_{\mathrm{DN}}}{1 - p_{\mathrm{DM}}} \\ M_{\mathrm{SCIT}} &= \frac{h_{\mathrm{G}}p_{\mathrm{I}}^{-1} + h_{\mathrm{V}}p_{\mathrm{I}}^{-1} + h_{\mathrm{I}} + h_{\mathrm{UMC}}p_{\mathrm{UM}}}{1 - p_{\mathrm{UM}}} \end{aligned}$$

As illustrated in Figure 7, MTTSF has a reciprocal relationship with the probability of intrusion, i.e., it decreases as the probability of intrusion rises. The proposed ITS architecture shows improved MTTSF with regard to $p_{\mathrm{I}}$ (17% compared to SITAR and 2% compared to SCIT) since it has more security features (e.g., proactive and reactive recovery) and thus more system states (corresponding to tolerance measures) when dealing with intrusions.

As shown in Figure 8, MTTSF ascends when the system spends more time in state G. In this graph, the proactive rejuvenation in SCIT seems to have more effects on the MTTSF when increasing the $h_{\mathrm{G}}$ in comparison with the reactive rejuvenation in SITAR. The acquired results show that the stability of our proposed ITS is better than the others. The improvement in MTTSF performance is 16% and 0.8% compared to SITAR and SCIT respectively. The acquired results for MTTSF also prove the security enhancement of the proposed architecture compared with the other two systems. As mentioned in availability analysis, the masking capabilities ($p_{\mathrm{M}}$) of the proposed ITS have been improved.

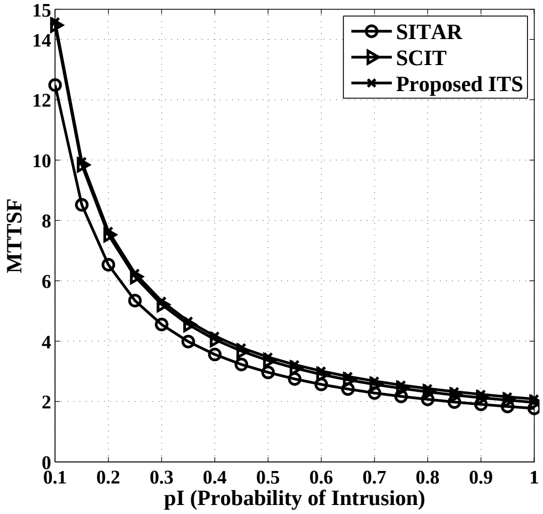


Figure 7: Impact of probability of intrusion on MTTSF

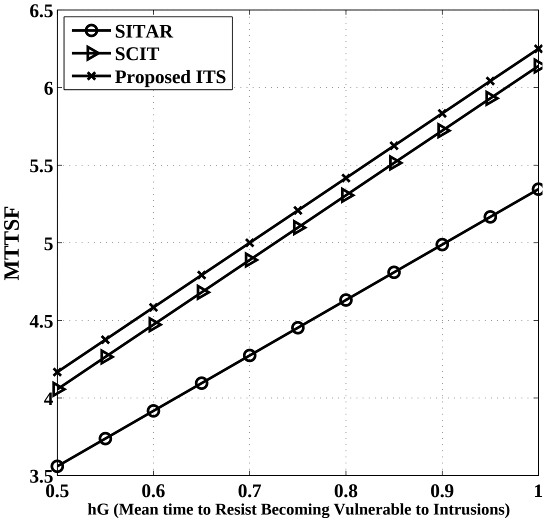


Figure 8: Impact of state holding time in state G on MTTSF

## 4.2 Cost Analysis

One of the downside of using intrusion tolerance is the substantial cost incurred by the underlying techniques such as redundancy and rejuvenation. Thus, the aforementioned techniques should be used meticulously. The proposed algorithms provide assistance for using redundancy and rejuvenation in an efficient manner to decrease the incurred cost. Since there is a trade-off between cost and security, the proposed algorithms make an effort to maintain an accepted level of security while reducing the associated cost. We considered two types of cost (in terms of overhead) as follows:

- Rejuvenation cost: The rejuvenation cost is represented by the number of system level rejuvenations.
- Redundancy cost: The redundancy level can be considered as a performance metric that can represent the incurred redundancy cost. It should be noted that the redundancy level is influenced by both proactive and reactive rejuvenation mechanisms since they affect the number of concurrent rejuvenations.

To demonstrate the efficiency of the proposed algorithms, a cost analysis is conducted using OMNeT++ simulator. We consider six different simulation scenarios as follows:

- $S_1$: Only system level proactive rejuvenation (as is the case in SCIT)
- $S_2$: Hierarchical proactive rejuvenation
- $S_3$: Only system level reactive rejuvenation (it is used as a part of reconfiguration measures in SITAR)
- $S_4$: Hierarchical reactive rejuvenation
- $S_5$: Hybrid system level rejuvenation (as is the case in Crutial)
- $S_6$: Hybrid and hierarchical rejuvenation (our proposed algorithms)

In all scenarios, the adaptive redundancy level algorithm is used. The primary factor that differentiates one scenario from another is the type of employed rejuvenation mechanisms. Moreover, it is assumed that the system is under sustained attack (i.e., the incoming traffic to the ITS always includes attack traffic as well). $S_6$ is the scenario that completely shows the proposed algorithms. $S_1$ to $S_6$ serve as proofs of concept for the impact of using different recovery mechanisms on the incurred cost and security of the system.

Other assumptions and simulation parameters are as follows. The number of replicas (whether active or under recovery), denoted by $RL$, at each moment during simulation is $2f + 1 + k$ with the minimum value of 4 (i.e., $f, k = 1$) and maximum value of 6 (i.e., $MaxRL$).The traffic distribution follows Poisson process. The time between arrivals are exponentially distributed with rate 2ms. It is assumed that with current parameters, no failure happens (i.e., meaning the attackers are not able to compromise more than $f$ replicas in a way that the infected replicas cannot be detected or the impact of the intrusion cannot be masked). So, the system will always be available and operates correctly. Moreover, the possible results will be calculated as the average over 10 runs except for the $S_1$ and $S_2$ in which no randomness is used and therefore repeating the experiment will not affect the results. The default parameters used in our simulation are tabulated in Table 3.

Table 3: Simulation assumptions

| Parameter | value |
|---|---|
| Number of runs | 10 |
| $f$ | 1 |
| $k$ | 1,2,3 |
| $MaxRL$ ($MaxDL$) | 6 |
| Redundancy level (RL) | 4-6 |
| Number of active replicas at the beginning of the simulation | 4 |
| Simulation time | 21600s |
| Number of critical processes in each replica | 5 |
| Number of backups for each process in per replica | 3 |
| Replica processing time | exponential(0.003s) |
| Process level reactive rejuvenation period | 300s |
| Process level proactive rejuvenation period | 240s |
| System level rejuvenation time | 600s |

### 4.2.1 Simulation Results and Discussion

This section demonstrates the efficacy of the proposed algorithms using different scenarios.

**Rejuvenation cost:** Figure 9 illustrates the number of performed system level rejuvenations in distinctive scenarios respectively. It is shown that the total number of carried out system level rejuvenations have been decreased in $S_6$ (in which our proposed hybrid and hierarchical recovery is used). Thus, the rejuvenation cost is reduced while the security and resiliency enhanced using a hybrid and hierarchical rejuvenation approach. Although in $S_2$ the number and time duration of recoveries have minimum value, the level of security and tolerance is not satisfactory because the system only uses hierarchical proactive recovery and it does not possess detection capability.

But as a virtue, $S_2$ shows the effectiveness of hierarchical proactive recovery in terms of rejuvenation cost reduction compared to $S_1$. This is also applicable to $S_4$ which demonstrates the desired effect of hierarchical reactive recovery compared to $S_3$ in which only system level reactive recovery is employed. Nevertheless, $S_4$ still suffers the problem of not being able to handle unknown and novel attacks as well as false positives. Also, $S_5$ (which uses hybrid system level rejuvenation) shows reduced rejuvenation overhead compared to $S_1$ and $S_3$ in which proactive and reactive system level recovery have been employed respectively.With regard to the aforementioned issues, we can draw the conclusion that the limitations of the first five scenarios are alleviated in $S_6$.

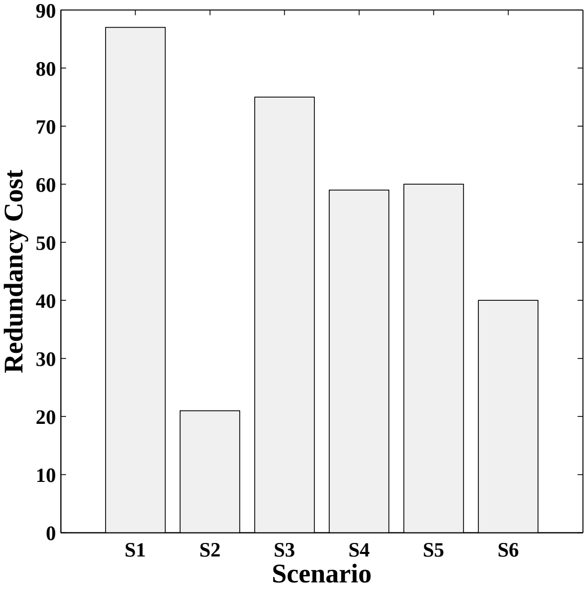


Figure 9: Rejuvenation cost in terms of the number of performed system level rejuvenations in each scenario

Figure 10 illustrates a detailed view of the provided results in Figure 9. In fact, the total number of rejuvenations in each scenario is gained from calculating the average values of these parameters for each replica over 10 simulation runs and then summing the averages (except for $S_1$ and $S_2$ as mentioned earlier). Furthermore, the redundancy level is reflected in Figure 10 by showing zero number of recoveries and zero rejuvenation time for the sixth replica in $S_1$ and $S_2$, i.e., the redundancy level is 5 ($RL = 5$) in these scenarios. This is mostly due to the inability of detection in these scenarios that would result in less number of performed concurrent rejuvenations and consequently less changes in the in the redundancy level.

**Process level and system level rejuvenations:** Figure 11 displays a view of the average number of performed

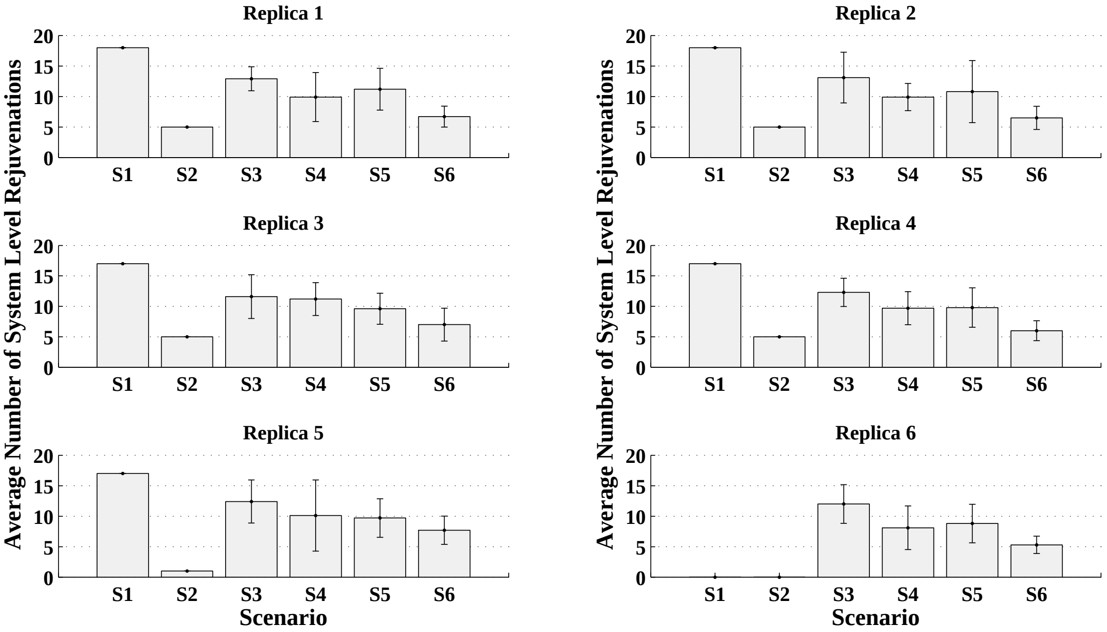


Figure 10: Average number of performed system level rejuvenations per replica for each scenario

process level rejuvenations and the average number of system level rejuvenations in scenarios that features hierarchical recovery, i.e., $S_2$, $S_4$ and $S_6$. In all three scenarios, most of the rejuvenations are carried out at process level. This shows the impact of hierarchical recovery on the reduction of recovery overhead resulting from system level rejuvenations (in $S_1$, $S_3$ and $S_5$ all the rejuvenations are done at system level). In $S_6$, the average number of process level rejuvenations has been increased compared to $S_4$ and consequently the average number of system level recoveries and rejuvenation overhead has been decreased. Although in $S_2$, the average number of system level rejuvenations is less than its peer in $S_6$, it does not satisfy the required level of security due to lack of detection abilities. Moreover, the average number of system level rejuvenations in $S_2$ depends on the frequency of triggering proactive recovery as well as the number of redundant processes related to each critical process in a replica. Assume there is a fixed number of redundant (backup) processes for each process. In this case, increasing the frequency of triggering proactive recoveries would result in the rise in the number of system level rejuvenations. In contrast, if the frequency of proactive recovery is constant, then increasing the number of backup processes assigned to each process, would reduce the number of system level recoveries (while increasing the number of process level recoveries). As it is shown in Table 3, the number of replicated processes is 4 for each crucial process in a replica. This means that it is expected to have on average of 4 process level rejuvenations per system level rejuvenation in $S_2$. This deterministic relation follows from the fact that $S_2$ only employs proactive recovery approach and it does not involve any randomness. Figure 11 confirms this matter for $S_2$. Although this relation is not deterministic for $S_4$ and $S_6$, with regard to Figure 11 we can roughly conclude that 2 and 3 process level recoveries are performed per system level recovery in $S_4$ and $S_6$ respectively. Obviously, $S_6$ shows less system level recovery cost in this case (more number of process level recoveries per system level recovery).

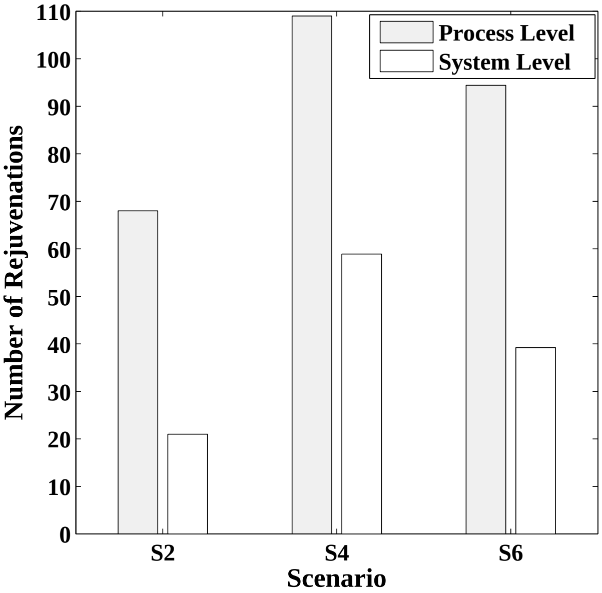


Figure 11: Average performed process level rejuvenations compared to system level rejuvenations

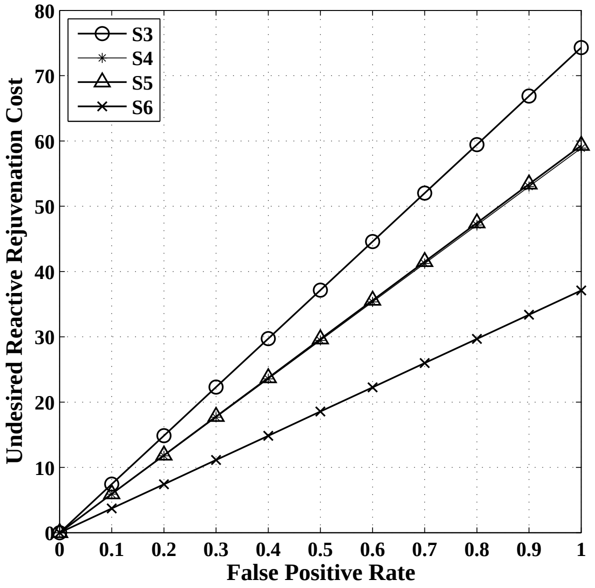


Figure 12: The impact of false positive rate on the undesired reactive recovery overhead

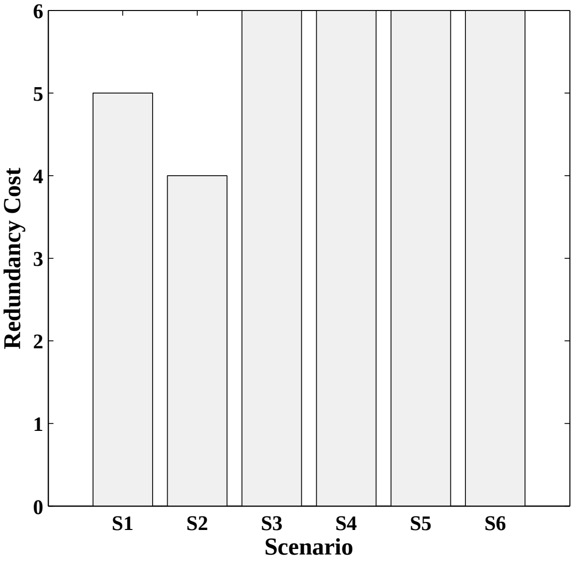


Figure 13: Average redundancy level at the end of simulation in each scenario

**Undesired reactive rejuvenation cost:** The false positive rate of the underlying detection mechanisms is an important factor that affects the reactive rejuvenation overhead. With respect to false positive and true positives, we can divide the reactive rejuvenation cost into two parts, namely desired reactive rejuvenation cost resulting from true positives and undesired reactive rejuvenation cost because of false positives. Therefore, as it is shown in Figure 12, the less the false positive rate is, the less the undesired rejuvenation cost would be in scenarios $S_3$-$S_6$. $S_1$ and $S_2$ are not shown in this figure since they only contain proactive recovery. $S_4$ and $S_5$ show almost the same behaviour since proactive rejuvenation in $S_5$ has minor impact on decreasing the number of system level reactive recoveries in this simulation; however, as mentioned earlier the security has been enhanced in $S_5$. Moreover, because the number of performed reactive recoveries in $S_3$-$S_5$ are more than $S_6$, the undesired reactive rejuvenation cost is increased sharply when the false positive rate rises. Thus, a fool-proof ITS is expected to have intrusion detection method with low false positive rate which subsequently result in less undesired reactive rejuvenation cost.

**Redundancy cost:** The average redundancy level at the end of simulation in different scenarios shows the effect of recovery mechanisms on the redundancy level (when the system is under attack). As shown in Figure 13, while the redundancy levels have the values of 5 and 4 for $S_1$ and $S_2$ respectively, the other four scenarios end up with level 6 which is the maximum value. Thus, in this simulation the effect of reactive recovery and hybrid recovery mechanisms on the redundancy level is more than the proactive recovery mechanism. When the system is armed with detection capabilities and reactive rejuvenation as well as proactive recovery, the possibility of increase in redundancy level is more (more attacks can be handled successful by the ITS). This follows from the fact that the ITS tries to adapt itself to the situation to maintain the desired availability. Thus, it increases the number of simultaneous rejuvenations which will have direct impact on the redundancy level.

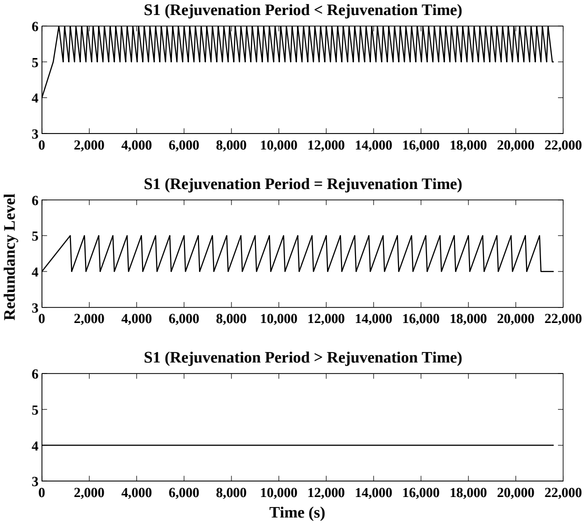


Figure 14: Proactive rejuvenation frequency impact on the redundancy level ($S_1$)

As a proof of concept, the impact of proactive recovery ($S_1$ and S2) on the redundancy level changes can be

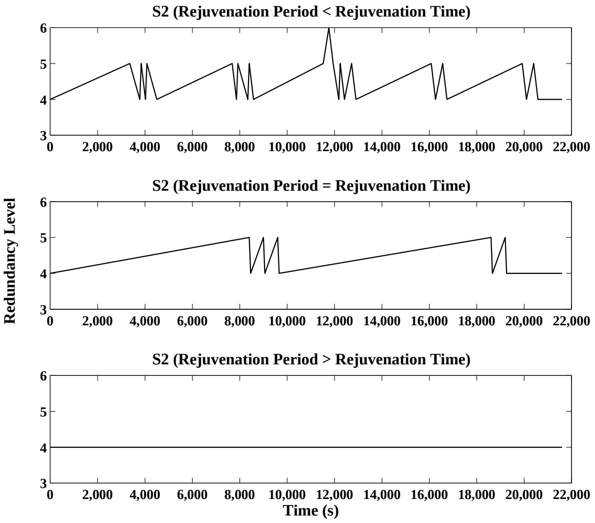


Figure 15: Proactive rejuvenation frequency impact on the redundancy level ($S_2$)

shown. As it has been depicted in Figures 14 and 15, the frequency of proactive rejuvenation has a direct impact on the redundancy level and subsequently on the number of concurrent system level rejuvenations. Three cases can be considered as follows:

1) Rejuvenation period < rejuvenation time: if this condition is met, then the probability of redundancy level changes is increased. This is due to the fact that most of the time the number of concurrent rejuvenations (i.e., k) is more than 1. That is, while one or more rejuvenations are in progress, another rejuvenation is triggered. In Figure 14, the number of redundancy level changes is more than its peer in Figure 15. This follows from the adoption of hierarchical proactive recovery in $S_2$ compared to $S_1$ that decreases the number of system level rejuvenations and the possibility of having more than one simultaneous rejuvenation. This also affects the maximum redundancy level during simulation time which is 6 for $S_1$ and 5 almost all the time for $S_2$. With regard to the aforementioned issues, the conclusion can be drawn that the redundancy cost has been declined in $S_2$.

2) Rejuvenation period = rejuvenation time: In this situation, the maximum redundancy level is 5 for both $S_1$ and $S_2$. Also, the shifts in redundancy level have been reduced for both scenarios. However, it is more tangible in $S_2$ which involves hierarchical rejuvenation. Obviously, $S_2$ shows less redundancy overhead compared to $S_1$. Concerning the incurred redundancy cost, this condition can be considered as a moderate situation. Nevertheless, if the time needed by an adversary to break into the system is less than the time between two successive recoveries (between two process level or system level recoveries or between a process level and a system level recovery) and the maximum redundancy level (i.e., 6) is reached, the required security level of the system will not be satisfied.

3) Rejuvenation period > rejuvenation time: Figures 14 and 15 illustrate no redundancy level change for $S_1$ and $S_2$ in this situation. So, the redundancy cost is minimum (the maximum redundancy level is 4 and the value of $k$ is always 1). In spite of this advantage, as mentioned before, the security requirements may not be satisfied (the intrinsic trade-off between security and cost).

# 5 Conclusion

Cyber security of critical infrastructures is a hot research area due to the fact that these systems are tightly coupled with ICT. The security incidents in cyber domain may affect the physical world and may subsequently lead to nationwide and disastrous consequences such as cascaded failures in the whole system. The possible ramifications would be financial loss or even loss of lives. Thus, this paper provided an in-depth research on the significance of using intrusion tolerance as a security approach to improve the security of critical infrastructures' control systems. An ITS architecture was proposed to be adopted in control centers' critical components. Using different intrusion tolerance techniques such as replication and diversity (along with dynamic redundancy level), hybrid and hierarchical recovery made the proposed ITS outperform two of well-known architectures, namely SITAR and SCIT. To investigate the effectiveness of the aforementioned features, analytical modelling and cost analysis were conducted. The acquired results demonstrated decrease in the incurred cost by intrusion tolerance techniques enhancement while maintaining an appropriate level of security.

# References

[1] N. Al Ebri, J. Baek, and C. Y. Yeun, "Study on secret sharing schemes (SSS) and their applications," in *International Conference for Internet Technology and Secured Transactions (ICITST'11)*, pp. 40–45, 2011.

[2] M. Badra and S. Zeadally, "An improved privacy solution for the smart grid," *International Journal of Network Security*, vol. 17, no. 4, pp. 1–8, 2015.

[3] A. K. Bangalore and A. K. Sood, "Securing Web servers using self cleansing intrusion tolerance (SCIT)," in *Second International Conference on Dependability (DEPEND'09)*, pp. 60–65, 2009.

[4] A. N. Bessani, "From Byzantine fault tolerance to intrusion tolerance (a position paper)," in *IEEE/IFIP*

*41st International Conference on Dependable Systems and Networks Workshops (DSN-W'11)*, pp. 15–18, 2011.

[5] A. N. Bessani, P. Sousa, M. Correia, N. F. Neves, and P. Verissimo, "The crutial way of critical infrastructure protection," *IEEE Security & Privacy*, vol. 6, no. 6, pp. 44–51, 2008.

[6] T. Y. Chang, M. S. Hwang, W. P. Yang, "An improved multi-stage secret sharing scheme based on the factorization problem," *Information Technology and Control*, vol. 40, no. 3, pp. 246–251, 2011.

[7] Y. Deswarte and D. Powell, "Internet security: An intrusion-tolerance approach," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 432–441, 2006.

[8] S. Distefano, F. Longo, and K. S. Trivedi, "Investigating dynamic reliability and availability through state–space models," *Computers & Mathematics with Applications*, vol. 64, no. 12, pp. 3701–3716, 2012.

[9] T. Dohi and T. Uemura, "An adaptive mode control algorithm of a scalable intrusion tolerant architecture," *Journal of Computer and System Sciences*, vol. 78, no. 6, pp. 1751–1774, 2012.

[10] M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro, "Analysis of operating system diversity for intrusion tolerance," *Software: Practice and Experience*, 2013.

[11] D. Ghosh, R. Sharman, H. R. Rao, and S. Upadhyaya, "Self-healing systems survey and synthesis," *Decision Support Systems*, vol. 42, no. 4, pp. 2164–2185, 2007.

[12] C. Griffin, B. Madan, and T. Trivedi, "State space approach to security quantification," in *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, vol. 2, pp. 83–88, 2005.

[13] B. E. Helvik, K. Sallhammar, and S. J. Knapskog, "8 Chapter - Integrated Dependability and Security Evaluation Using Game Theory and Markov Models," in *Information Assurance*, pp. 209–245, Morgan Kaufmann, Burlington, 2008.

[14] J. H. Huang and F.-F. Wang, "The strategy of proactive-reactive intrusion tolerance recovery based on hierarchical model," in *Web Information Systems and Mining*, vol. 6987 of *Lecture Notes in Computer Science*, pp. 283–293, 2011.

[15] P. Kabiri and A. A. Ghorbani, "Research on intrusion detection and response: A survey," *International Journal of Network Security*, vol. 1, no. 2, pp. 84–102, 2005.

[16] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *37th Annual Conference on IEEE Industrial Electronics Society (IECON'11)*, pp. 4490–4494, 2011.

[17] J. C. Knight, J. Hill, P. Varner, A. L. Wolf, D. Heimbigner, and P. Devanbu, "Willow system demonstration," in *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX'03)*, vol. 2, pp. 123–125, 2003.

[18] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: Cyber attacks, countermeasures, and challenges," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 38–45, 2012.

[19] B. B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," *Performance Evaluation*, vol. 56, no. 1-4, pp. 167–186, 2004.

[20] A. Nagarajan and A. Sood, "SCIT and IDS architectures for reduced data ex-filtration," in *International Conference on Dependable Systems and Networks Workshops (DSN-W'10)*, pp. 164–169, 2010.

[21] Q. Nguyen and A. Sood, "Quantitative approach to tuning of a time-based intrusion-tolerant system architecture," in *3rd Workshop Recent Advances on Intrusion-Tolerant Systems (WRAITS'09)*, pp. 132–139, 2009.

[22] Q. L. Nguyen and A. Sood, "A comparison of intrusion-tolerant system architectures," *IEEE Security and Privacy*, vol. 9, no. 4, pp. 24–31, 2011.

[23] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "SCADA security in the light of cyberwarfare," *Computers & Security*, vol. 31, no. 4, pp. 418–436, 2012.

[24] D. Niyato, P. Wang, and E. Hossain, "Reliability analysis and redundancy design of smart grid wireless communications system for demand side management," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 38–46, 2012.

[25] J. Pieprzyk and X. M. Zhang, "Ideal secret sharing schemes from permutations," *International Journal of Network Security*, vol. 2, no. 3, pp. 238–244, 2006.

[26] S. B. E. Raj and G. Varghese, "Analysis of intrusion-tolerant architectures for web servers," in *International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT'11)*, pp. 998–1003, 2011.

[27] A. Saidane, V. Nicomette, and Y. Deswarte, "The design of a generic intrusion-tolerant architecture for web servers," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 1, pp. 45–58, 2009.

[28] J. T. Seo and C. Lee, "The green defenders," *IEEE Power and Energy Magazine*, vol. 9, no. 1, pp. 82–90, 2011.

[29] sKyWIper Analysis Team, "sKyWlper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks," tech. rep., Laboratory of Cryptography and System Security (CrySyS Lab), 2012.

[30] P. Sousa, A. N. Bessani, and R. R. Obelheiro, "The FOREVER service for fault/intrusion removal," in *Proceedings of the 2nd workshop on Recent advances on intrusiton-tolerant systems (WRAITS;08)*, pp. 1–6, New York, USA, 2008.

[31] P. Sousa, A. N. Bessani, M. Correia, and N. F. Neves, "Highly available intrusion-tolerant services with proactive-reactive recovery," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 452–465, 2010.

[32] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber physical system security for the electric power grid," *Proceedings of the IEEE*, vol. 100, pp. 210–224, Jan. 2012.

[33] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[34] R. Stroud, I. Welch, J. Warne, and P. Ryan, "A qualitative analysis of the intrusion-tolerance capabilities of the MAFTIA architecture," in *International Conference on Dependable Systems and Networks (DSN'04)*, pp. 453–461, 2004.

[35] M. Tanha and F. Hashim, "An intrusion tolerant system for improving availability in smart grid control centers," in *2012 18th IEEE International Conference on Networks (ICON'12)*, pp. 434–440, 2012.

[36] M. Tanha and F. Hashim, "Towards a secure and available smart grid using intrusion tolerance," in *Internet and Distributed Computing Systems*, Lecture Notes in Computer Science, pp. 188–201, 2012.

[37] C. W. Ten, G. Manimaran, and C. C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 4, pp. 853–865, 2010.

[38] T. Uemura, T. Dohi, and N. Kaio, "Availability analysis of an intrusion tolerant distributed server system with preventive maintenance," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 18–29, 2010.

[39] A. Valdes, M. Almgren, S. Cheung, Y. Deswarte, B. Dutertre, J. Levy, H. Saidi, V. Stavridou, and T. E. Uribe, "Dependable intrusion tolerance: technology demo," in *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX'03)*, vol. 2, pp. 128–130 vol.2, 2003.

[40] P. E. Verissimo, N. F. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud, and I. Welch, "Intrusion-tolerant middleware: the road to automatic security," *IEEE Security & Privacy*, vol. 4, no. 4, pp. 54–62, 2006.

[41] P. Veríssimo, N. Neves, and M. Correia, "Intrusion-tolerant architectures: Concepts and design," in *Architecting Dependable Systems*, vol. 2677 of *Lecture Notes in Computer Science*, pp. 3–36, 2003.

[42] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient Byzantine fault-tolerance," *IEEE Transactions on Computers*, vol. 62, no. 1, pp. 16–30, 2013.

[43] F. Wang, R. Uppalli, and C. Killian, "Analysis of techniques for building intrusion tolerant server systems," in *IEEE Military Communications Conference (MILCOM'03)*, vol. 2, pp. 729–734 Vol.2, 2003.

[44] F. Wang, F. Jou, F. Gong, C. Sargor, K. Goseva-Popstojanova, and K. Trivedi, "SITAR: a scalable intrusion-tolerant architecture for distributed services," in *Foundations of Intrusion Tolerant Systems [Organically Assured and Survivable Information Systems]*, pp. 359–367, 2003.

[45] R. Wang, F. Wang, and G. T. Byrd, "Design and implementation of acceptance monitor for building intrusion tolerant systems," *Software - Practice and Experience*, vol. 33, no. 14, pp. 1399–1417, 2003.

[46] Y. S. Wang and L. Wang, "Secure server switching system," in *Computer Engineering and Applications (ICCEA'10), 2010 Second International Conference on*, vol. 1, pp. 224–228, Mar. 2010.

[47] F. Zhao, M. Li, W. Qiang, H. Jin, D. Zou, and Q. Zhang, "Proactive recovery approach for intrusion tolerance with dynamic configuration of physical and virtual replicas," *Security and Communication Networks*, vol. 5, no. 10, pp. 1169–1180, 2012.

[48] L. Zhou, F. B. Schneider, and R. Van Renesse, "COCA: A secure distributed online certification authority," *ACM Transactions on Computer Systems,*, vol. 20, pp. 329–368, Nov. 2002.

**Maryam Tanha** is currently perusing her PhD in the Department of Computer Science, Faculty of Engineering, University of Victoria. She received her M.Sc. in Communication and Network Engineering from Universiti Putra Malaysia in 2013. Her research activities are focused on Software Defined Networking and survivability analysis, particularly for critical infrastructures. She is a member of IEEE.

**Fazirulhisyam Hashim** holds a M.Sc. degree from the Universiti Sains Malaysia and a Ph.D. degree from the University of Sydney, Australia. He is currently a researcher and senior lecturer at the Wireless and Photonic Network Research Center of Excellence (WiPNET) at the Universiti Putra Malaysia. His research interests include network security and QoS of next generation mobile networks, green communication systems, and wireless sensor networks. He is a member of IEEE.

**Shamala K. Subramaniam** received her B.Sc. degree in Computer Science from Universiti Putra Malaysia, in 1996, M.S. (UPM), in 1999 and PhD (UPM) in 2002. She is currently an Associate Professor in the Department of Communication Technology and Networking, Faculty of Computer Science, Universiti Putra Malaysia. Her research interests are computer networks, simulation and modelling, scheduling and real time systems.

# Ranking Intrusion Likelihoods with Exploitability of Network Vulnerabilities in a Large-Scale Attack Model

Rattikorn Hewett[1] and Phongphun Kijsanayothin[2]
*(Corresponding author: Rattikorn Hewett)*

Department of Computer Science[1]
Texas Tech University, Lubbock, Texas, USA
Department of Electrical and Computer Engineering[2]
Naresuan University, Phitsanulok, Thailand
(Email: rattikorn.hewett@ttu.edu)

*(Received Oct. 30, 2013; revised and accepted July 24, 2014)*

## Abstract

Network vulnerabilities are common sources of many security threats. Attack models representing chains of all possible vulnerability exploits by attackers can help locate security flaws and pre-determine appropriate preventative measures. To realize the full benefits of attack models, effective analysis is crucial. However, due to the size and complexity of the models, manually pinpointing potential critical attacks can be daunting. Thus, there is a need for an automated analysis approach. Existing techniques are either based on network topology alone or subjective prior knowledge. They do not utilize domain-specific knowledge. This paper presents an approach to automatically ranking states in an attack model in the order of their intrusion likelihoods. Using the degree of exploitability of network vulnerabilities and the Markov property, the proposed approach provides a tractable computation enhanced by domain-specific heuristic knowledge for estimating such likelihoods. The paper discusses the details of the approach, illustrates its use, and compares results with a similar existing technique with experiments on its performance.

*Keywords: Attack graphs, security models, network vulnerability, network security, ranking algorithm*

## 1 Introduction

Securing networks requires understanding of *network vulnerabilities*, which are common sources of many attacks. Such vulnerabilities include exploitable errors in configurations (e.g., ports and services enabled) or the network service software (e.g., *Apache Chunked-Code* on Apache web servers, *buffer overflow* on Windows XP SP2 operating environments, and *TNS-Listener* on Oracle software for database servers). These vulnerabilities are unavoidable as long as we need the network to provide their corresponding services. Building attack models as chains of all possible *vulnerability exploits* by attackers can help security administrators locate security flaws and pre-determine appropriate preventative measures. To fully realize the practicality of attack models, effective analysis is crucial. By analysis, we mean a systematic method for extracting useful information for security management.

Much work in attack model analysis has been primarily on visualization [9, 16, 17, 22]. Although this can help security administrators assess overall threats to the network, locating hazardous situations and locations to secure networks is still a challenging task due to the size and complexity of the attack models. Besides, visualization often requires human expertise to observe and pinpoint critical information. Thus, visualization can be time consuming and may produce inconsistent findings. There is a need for an automated approach to attack model analysis that can assess network security more effectively.

Several formal approaches to automatic attack model analysis have been proposed using graph theory [10], probabilistic analysis [21] and game theory [13]. The probabilistic analysis by Sheyner et al. [21] estimates the reliability of a given node (or state) in the attack model (or attack graph) in term of the probability of an attacker reaching his goal from the node. However, the assignment of arbitrary prior probabilities of detecting each attack action makes this approach ad-hoc as it relies on subjective opinions. Jha et al. [10] introduced a graph-based approach that identifies the smallest set of exploits to be removed to prevent the network from all possible attacks shown in a given attack model. The intent is to identify the smallest set of counter-measures required to protect the network. However, the choice of an appropriate set of counter-measures does not always depend on its size alone. Furthermore, the approach is limited to directed acyclic attack graphs (DAG). Lye and Wing [13] applies a game theoretic approach to model "rational" interactions between an attacker and a network administrator during an attack attempt. Unlike others, this approach is not a preventative approach and its application is restricted to complete attack graphs.

A recent approach to attack model analysis aims to efficiently rank the nodes of an attack model based on the likelihood of an attacker reaching these states was introduced by Mehta et al. [14]. The ranking provides useful information for determining which attack path is more vulnerable or requires more immediate attention for network protection. The approach is based on *PageRank* [4], a well-known link analysis algorithm for Google's web search engine. Unfortunately, their ranking results are not always meaningful. This is because network intrusion does not have as much freedom as web browsing where we can randomly visit any website via URLs. In network intrusion, an attacker can only advance his attack position to a node that has connectivity and vulnerability to be exploited. Thus, the approach to computing the probability of advancing each attack action to a new state requires an adjustment. Mehta el al. introduced a modified ranking algorithm to address this issue. However, all of the above approaches tend to view attack model analysis as a general problem in graph theory and only use structural topology of the attack model. None makes use of domain-specific knowledge about network security (e.g., vulnerability and degree of its exploitability) to obtain more meaningful and accurate analysis.

This paper presents an approach to automatically analyzing security attack models that ranks states in the attack model in the order of their likelihoods of being intruded by an attacker. The proposed approach is most similar to Mehta et al.'s approach [14]. However, there are a few major differences that set this work apart from previous work. First, we use knowledge about the *exploitability* of network vulnerability instead of subjective or no prior domain-specific knowledge in estimating the intrusion likelihoods as in Mehta et al.'s approach. In particular, our analysis proposes *ExploitRank*, a new heuristic ranking algorithm that uses public information on the *Common Vulnerability Scoring System* [7] as a measure for quantifying the exploitability of network vulnerability. Second, *ExploitRank* assumes that when an attacker has no more vulnerability to exploit to advance to the next state, he will give up on the current path and start finding an alternative attack path from the beginning (i.e., at initial states). In contrast, Mehta el al.'s approach assumes that an attacker may either persist on attacking the same state (analogous to browsing a web page that has links to itself) or decide to start over. We will show that these slight differences yield drastically different results and that our approach produces results that better match logics in our reasoning than those of Mehta et al.'s. The paper has the following contributions:

1) An automated framework for protecting a computer network against malicious attacks via attack models.

2) An enhanced ranking algorithm for analyzing large-scale attack models by ranking possible attack states based on their relative intrusion likelihoods.

The ExploitRank algorithm help provide priorities for network security management. The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives preliminary concepts. Sections 4 and 5 describe and illustrate our proposed approach with some experimental results. Section 7 concludes the paper.

## 2 Related Work

Majority of research in attack model analysis includes visualization techniques [16, 17] that have been employed to simplify an attack model. By grouping together nodes that have the same characteristics (e.g., same locality) into a single node, the resulting model is easier to view and less complicated to find ways to prevent attack paths. Because most graph visualization is semi-automated or manual, it tends to be time consuming and the results obtained can still be far too complex to be useful in practice. Our approach is automated and does not aim to simplify the view of the model but helps locate critical nodes.

Sheyner et al.'s probabilistic approach [21] employs Markov Decision Process (MDP) to estimate reliability of each node in the model with the probability of an attacker reaching his goal from a given node. This approach tends to be subjective and impractical since it requires assignments of arbitrary prior probabilities of detecting each attack action. Our approach, however, does not require such prior probability assignment.

A graph-based approach by Jha et al. [10] aims at finding a minimum set of countermeasures to guarantee that the attackers' goal states will never be reached. This is done by estimating the smallest set of attacks required to protect the network along with the smallest set of countermeasures to account for each of the attacks. Jha et al.'s approach is limited to a DAG, where each attack path, from an initial state to a goal state, is considered only once, whereas our approach can be applied to any attack graph topology.

Another approach to attack graph analysis aims at ranking graph states by their likelihoods of being attacked [14, 19]. Most of these ranking techniques are based on the well-known PageRank algorithm [4] for ranking web pages. Among these, the work that is most closely related to our approach is Mehta et al.'s approach [14] that modifies the transitions at the end of each attack path to an initial state instead of every node as used in PageRank algorithm. However, Mehta el al. treat each node reachable from a given node to have the same degree of vulnerability and exploitability. Unlike ours, none of the above approaches exploits domain-specific knowledge about the exploitability of the network vulnerabilities.

# 3 Preliminaries

## 3.1 Terms and Concepts in Network Security

*Network vulnerabilities* refer to the weaknesses of a target system network, for examples, security flaws in server software (e.g., *Apache Chunked-Code*, *Oracle with TNS Listener* software) or network configurations (e.g., enabled ports and services). Known vulnerabilities are publicly available (e.g., [5, 18]). Vulnerability can be exploited when its preconditions are satisfied. These preconditions include connectivity, access privileges on relevant hosts, and network or host configurations.

A *vulnerability exploit* refers to an attacker's *action* to advance his attack. Typically, an exploit involves an *attacking host* (the source on which an attacker performs an exploit), and a *victim host* (the destination on which an attacker gains benefits after the exploit has been carried out). An exploit has two modes: *local* and *remote.* To attack, the network must have vulnerabilities and an attacker must know how to exploit them. Note that each exploit could involve one or more vulnerabilities (e.g., the "Apache Chunked-Code Buffer Overflow" exploit involves software vulnerability (e.g., Apache web server software Version 1.3) and configuration vulnerability (e.g., Apaches default port is enabled on a victim host). Similarly, vulnerability could be involved in more than one exploit.

An *attack model* or *attack graph* represents the behavior of attackers harming a network. Each node in the graph represents a state, typically specified by the relevant network attributes such as connectivity between hosts and an attacker's access privileges. Each link represents an action that an attacker takes to gain his access control in the network. Starting from a set of initial nodes, an attacker can take an action that exploits the network vulnerability to reach a set of states satisfying the attacker goal (e.g., obtain a root privilege on a database server). There are various forms of attack graphs (e.g., access [1], host-centric [8], and network-based [21]). However, they all use the same level of abstraction of attacker's actions. Each attack model can have multiple initial states as well as multiple goal states.

## 3.2 Link Analysis and the PageRank Algorithm

Ranking web pages is an important function of an Internet search engine. Approaches to ranking web pages are based on a link analysis, where we assign weights to a hyperlinked set of web pages to approximate the relative importance of each web page within the set. Variations of link-based ranking algorithms include *PageRank* [4] and *HITS* [12]. Because of its accuracy and efficiency, the Google's *PageRank* algorithm becomes one of the most predominant ranking algorithms, whose main concepts will be briefly described below.

The rank value of a web page indicates a probability that a web surfer randomly clicking on links will end up visiting the page. Thus, *the sum of page rank values over all of the considered web pages must be one.* It is assumed that the initial approximation of this probability would be equally distributed among all web pages in the considered collection. *PageRank* algorithm simulates the clicking behavior of a web surfer who can visit a web page either via an incoming link to the page or picking a URL of the page at random. The surfer who randomly clicks on links will eventually stop. At any surfing stage, a *damping factor* is the probability that the web surfer will continue surfing using hyperlinks.

Let $r_t(v)$ be the probability of visiting web page $v$ at the time $t$, $d$ be a damping factor and $V$ be a set of web pages under consideration. For a page $v$, $out(v)$ and $in(v)$ is a set of web pages in $V$ with an outgoing link from $v$, and an incoming link to $v$, respectively. The page rank value is recursively defined and its computation can be viewed as a Markov process whose state are pages and the links between pages represent state transitions that are equal probable. The *PageRank*'s computation is given in the Equation (1) below.

$$r_{t+1}(v) = (1-d)\sum_{u \in V} \frac{r_t(u)}{|V|} + d \sum_{u \in in(v)} \frac{r_t(u)}{|out(u)|} \qquad (1)$$

The second part of Equation (1) represents when the surfer continues surfing (with probability $d$) to page $v$ at time $t+1$ by clicking a hyperlink, at time $t$, from each page $u$ that has an outgoing link to $v$ (i.e., $u \in in(v)$). Because the chance of clicking each of such page $u$ is equally likely, the probability of visiting v from each such $u$ is $1/|out(u)|$, assuming that $u$ has no more than one link to $v$. Alternatively, the surfer may stop using hyperlinks (with probability $1-d$) but visit page $v$ at time $t+1$ by using page $v$'s URL from any page that the user is at time $t$. The probability of visiting $v$ via URL from any page is $1/|V|$ and thus, we obtain the first part of the equation.

To satisfy the constraint that the sum of page rank values over all of the considered web pages at any time must be one, a web page that has no outgoing hyperlink is assumed to have a link pointing to itself. To see this, consider summing $r_{t+1}(v)$, from Equation (1), over all $v$. Thus, the left side yields one by the constraint. On the right side of the equation, the first part gives $(1-d)$ and the second part becomes:

$$d \sum_{u \in V} \sum_{u \in in(v)} \frac{r_t(u)}{|out(u)|} = d \sum_{|out(u)| \neq 0} r_t(u). \qquad (2)$$

The second part as shown in Equation (2) needs an additional term, $d\sum_{|out(u)|=0} r_t(u)$, to produce a total of

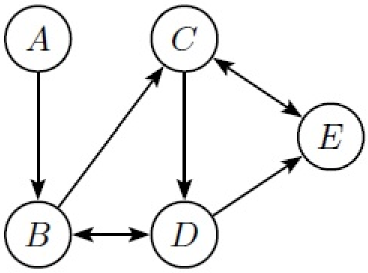


Figure 1: Hyperlinks of Web pages

$d$ in the second part of the equation so that the right side of the equation becomes one as desired. Thus, the extra term is required for the constraint to hold. In fact, adding this extra term is acquired by assuming for a page with no outgoing link to have a self-loop link. With this assumption, the constraint is satisfied. As a result, Equation (1) can be simplified as the following.

$$r_{t+1}(v) = \frac{(1-d)}{|V|} + d \sum_{u \in in(v)} \frac{r_t(u)}{|out(u)|} \tag{3}$$

The above computation iterates over time to obtain a stable estimate of the probability distribution of each page's visit by random clicking behaviors. Thus, the computation terminates when there is no change in the probability distribution obtained.

We now give a small example to illustrate how the *PageRank* algorithm works. Figure 1 shows a collection of five web pages where a hyperlink between the pages is represented by a directed edge.

Based on the above web page structure, we can create a *stochastic matrix* (or *transition matrix*) [2] $A = (a_{ij})$, where $a_{ij}$ represents a probability that a surfer makes a transition from page $i$ to page $j$. Here we assume that each outgoing page from the same page has equal chance to be visited. Thus, below is a stochastic matrix corresponding to the hyperlinks of web pages in Figure 1. Here $B$ has two outgoing links to page $C$ and $D$. Thus, $a_{BC} = a_{BD} = 1/2$. Stochastic matrix is used for computing transitions in each iteration step in a Markov process.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Initially at $t = 0$, each of the five pages has the same ranking value of 1/5 because the probability of visiting each page is equally likely and the sum of these probabilities must be one. Using a commonly used value of $d = 0.85$, at $t = 1$, the ranking values of pages $A$, $B$, $C$, $D$ and $E$ obtained are 0.03, 0.285, 0.115, 0.200, and 0.370, respectively. Table 1 summarizes the results up until iteration 15, where the ranking values no longer change.

Table 1: Ranking results of *PageRank* algorithm

| $t$ | $r_t(A)$ | $r_t(B)$ | $r_t(C)$ | $r_t(D)$ | $r_t(E)$ |
|---|---|---|---|---|---|
| 0 | 0.200 | 0.200 | 0.200 | 0.200 | 0.200 |
| 1 | 0.030 | 0.285 | 0.115 | 0.200 | 0.370 |
| 2 | 0.030 | 0.141 | 0.151 | 0.200 | 0.478 |
| ... | ... | ... | ... | ... | ... |
| 14 | 0.030 | 0.099 | 0.072 | 0.103 | 0.696 |
| 15 | 0.030 | 0.099 | 0.072 | 0.103 | 0.696 |

As shown in the last column of Table 1, page $E$ has the highest value of 0.696 and therefore it has the highest chance to be visited comparing to other pages. In fact, the ranking order of these web pages, based on their visit likelihoods, is $E$, $D$, $B$, $C$, $A$, respectively.

# 4 Proposed Approach

The proposed analysis applies domain-specific knowledge to estimate the probability distribution of intrusion for each attack state in a given attack model. Specifically, it identifies, for each attack state, a relative chance of intrusion based on the degree of exploitability of its vulnerabilities. This section describes two core components of our analysis approach. Section 4.1 defines exploitability as heuristics to be applied in the *ExploitRank* algorithm, which is to be described in Section 4.2.

## 4.1 Exploitability

Our approach uses knowledge about existing network vulnerabilities that can be found in public databases. It is well recognized that some vulnerability may be exploited more easily than others. In fact, the complexity of the vulnerability can affect its exploitability, which in turn influences the chance of intrusion at different states of the network attack.

We define *exploitability*($v$) to be a function that measures a degree of difficulty in exploiting vulnerability $v$ with values ranging from zero to one (i.e., from the hardest to the easiest to exploit, or from the lowest to the highest vulnerability). The Common Vulnerability Scoring System (CVSS) [7] and severity factor provides a standard for computing exploitability of various publicly known vulnerabilities. Basic CVSS is based on vulnerability characteristics that are static over time and user environments. There are three basic metrics: *access vector*, *access complexity*, and *authentication.*

*Access vector* represents difficulty from the access location (e.g., local, adjacent network accessible, and network accessible or remote) required to exploit the vulnerability. The more remotely an attacker can exploit the vulnerability, the greater the exploitability value will be. *Ac-*

*cess complexity* indicates the level (i.e., low, medium and high) of effort required to exploit the vulnerability after an access to the target point is gained. For example, a buffer overflow in an Internet server has low complexity since the vulnerability can be exploited once an attacker gains access of the server. The lower the complexity is, the higher the exploitability will be. Finally, *authentication* is defined to measure the number of authentications required (e.g., multiple instances, single instance, or no instance) before network vulnerability can be exploited. Based on the United State National Institute of Standard and Technology [15], qualitative domain values of these three CVSS metrics are quantified to numeric values as the following:

$$\begin{aligned}
\textit{Access Vector} &= \text{case } \textit{Access Vector} \text{ of}\\
&\phantom{=}\ \text{Local access: } 0.395\\
&\phantom{=}\ \text{Adjacent network accessible: } 0.646\\
&\phantom{=}\ \text{Network accessible: } 1.0\\
\textit{Access Complexity} &= \text{case } \textit{Access Complexity} \text{ of}\\
&\phantom{=}\ \text{High: } 0.35\\
&\phantom{=}\ \text{Medium: } 0.61\\
&\phantom{=}\ \text{Low: } 0.71\\
\textit{Authentication} &= \text{case } \textit{Authentication} \text{ of}\\
&\phantom{=}\ \text{Multiple instances : } 0.45\\
&\phantom{=}\ \text{Single instance : } 0.56\\
&\phantom{=}\ \text{No authentication: } 0.704
\end{aligned}$$

As an example, consider CVE-2006-5794, the Common Vulnerabilities and Exposures (CVE) in the sshd Privilege Separation Monitor in OpenSSH before Version 4.5. This vulnerability can be exploited by network accessible (i.e., remote) with no authentication, and the method to exploit this vulnerability is of low complexity. Therefore, *access vector*, *authentication*, and *access complexity* of this vulnerability is 1.0, 0.704 and 0.71, respectively. Thus, given a severity factor of 20, the exploitability of the CVE-2006-5794 vulnerability can be calculated as $20 \times AccessVector \times AccessComplexity \times Aunthentication = 9.9968$. Note that the exploitability of the vulnerability in [3, 8] has a maximum value of 10. To obtain the exploitability value ranging to a maximum of one as defined in this paper, we divide the resulting value by 10. This gives the *exploitability*(CVE-2006-5794) = 0.99968, which indicates that CVE-2006-5794 has a high exploitability degree and thus, high vulnerability (i.e., easy to exploit/attack).

## 4.2 The *ExploitRank* Algorithm

*ExploitRank* algorithm estimates the probability distribution of intrusion for each attack state in a given attack model by applying Markov model similarly to how *PageRank* algorithm applies the model for ranking web pages. However, there is a subtle difference between web surfing behaviors and network attacking behaviors.

While a web surfer can randomly pick a web page to visit via its URL, an attacker does not have the same freedom. In fact, an attack model provides a constraint of how an attacker can traverse among attack states. For example, a surfer can arrive at any web page in one single step via URL but an attacker requires more than one step to advance to an attack state that the target system is completely shut down (e.g., by first gaining access privilege of the target system followed by a few steps to exploit the target's vulnerability). For this reason, we cannot employ the same recurrences of Equations (2) and (3) for ranking exploitability in attack states.

During an attack, an attacker has options to *continue* or *quit* attacking on a current path. We assume that if the attacker quits attacking on the current path (because it is too hard to lead to his goal), he will attempt on an alternative path by starting over from one of the set of initial states. Each of the initial state has equal chance to be a starting point of this new attempt. On the other hand, if he continues attacking, he will advance to each of the possible transition states with a *probability* based on how hard its vulnerabilities can be exploited (see more details later).

Based on a Markov model, we obtain Equation (4) and Equation (5) for computing the probability distribution of intrusion of a given attack model where we use the exploitability of vulnerabilities at each attack state along with the structure of the network. The computation gives a relative chance of intrusion for each attack state, or, roughly speaking, a ranking of the exploitability of attack states in the attack model. Thus, it provides a basis for the proposed *ExploitRank* algorithm.

For a given security model, let $r_t(v)$ be the probability of intrusion of attack state $v$ at time $t$, $I$ be a set of initial states and $h(u, v)$ be the exploitability of a vulnerability exploit from $u$ to $v$ as explained in Section 4.1. We define $r_t(v)$, a ranking score of $v$ at time $t$, recursively as follows:

**Case 1:** $v$ is not an initial state

$$r_{t+1}(v) = \sum_{u \in in(v)} r_t(u) \cdot e(u, v) \tag{4}$$

**Case 2:** $v$ is an initial state

$$\begin{aligned}
r_{t+1}(v) \quad &= \quad \sum_{u \in in(v)} r_t(u) \cdot e(u, v) \quad + \\
&\frac{1}{|I|}\left( \sum_{\substack{u \in V \\ w \in out(u)}} r_t(u) \cdot \bar{e}(u, w) + \sum_{\substack{u \in V \\ out(u) = \emptyset}} r_t(u) \right)
\end{aligned} \tag{5}$$

where $e(u, v) = \frac{h(u,v)}{|out(u)|}$ and $\bar{e}(u, v) = \frac{1-h(u,v)}{|out(u)|}$.

When $v$ is not an initial state, the only way to attack $v$ is by continuing exploiting a vulnerability from any state $u$ to $v$, where $u$ was attacked in a previous step, i.e., $u \in in(v)$. The likelihood of attack from each such $u$ depends on the chance to intrude $u$, i.e., $r_t(u)$, and the likelihood of each vulnerability exploit from $u$ to $v$, i.e., $e(u, v)$. The latter depends on the chance of selecting the move from $u$ to $v$ out of all possible moves from $u$, i.e., $1/|out(u)|$ and the probability based on how hard it is to apply the exploit, i.e., $h(u, v)$, the exploitability of the vulnerability exploit from $u$ to $v$. Thus, we obtain Equation (4).

On the other hand, when $v$ is an initial state, an attacker can reach $v$ in two ways. First, by continuing advancement from previously intruded state as derived in Equation (4), we can obtain the first part of Equation (5). The other way to reach $v$ is based on our assumption that when the attacker gives up on the current attack path, he will start over from any initial state. Thus, the likelihood to intrude each initial state $v$ depends on the chance to intrude any possible state $u$, i.e., $r_t(u)$, and the chance that the attacker will not continue exploiting a vulnerability from $u$ to start over from $v$. If $u$ has a vulnerability exploit to $w$, i.e., $w \in out(u)$ and $u$ is not a terminal node, then the chance of $u$ not to continue with this exploit and start over at $v$ (out of all possible initial states) is $\frac{1}{|I|} \times \frac{1-h(u,w)}{out(u)}$. However, if $u$ is a terminal node, $u$ does not have an out-going exploit and by our assumption, the chance of the attacker not to exploit the vulnerability from $u$ is certain. Thus, the chance of $u$ starting over at $v$ becomes $1/|I|$. This gives Equation (5).

Based on the recurrence equation above, we construct the *ExploitRank* algorithm as shown in Algorithm 1, where all variables are as defined. Assume that any given attack model can be represented as a graph, $G(V, E)$, where $V$ and $E$ represents a set of attack states and a set of vulnerability exploits, respectively. *ExploitRank* takes the attack model $G$ with an exploitability degree corresponding for each possible connection between attack states as inputs. The probability distribution of network intrusion is computed recursively and iteratively using the stochastic matrix, defined in line 22, until the process reaches a stationary point in line 31. The algorithm produces a *relative* chance of intrusion at each attack state in a given model. This can be viewed as ranking among attack states in the order of the exploitability of their vulnerabilities. Next we evaluate the proposed approach by comparing the results obtained from the ranking between with and without the proposed heuristic (i.e., ours vs. Mehta et al.'s approach).

Note that Mehta et al. adopted the assumption used in *PageRank* algorithm where the attacker (or web surfer) may still pursue attacking (surfing) the terminal state $u$ with probability $d$, the damping factor, leaving the chance of attacking $v$ to be $1 - d$. This difference with our approach is shown in Figure 2, where $A$ is an initial node. In addition, Mehta et al.'s approach does not provide an explicit formulation of the Markov model as expressed in the equations here. More importantly, their approach does not take the degree of the difficulty in exploiting the vulnerability into consideration.

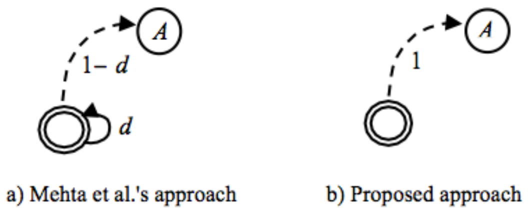
**Algorithm 1** ExploitRank

```
1: Procedure ExploitRank$G(V, E), h$
2: $I \leftarrow asetofinitialstates$
3: $A \leftarrow zeromatrixofsize|V| \times |V|$
4: $t \leftarrow 0$
5: for each $v \in V$ do
6:    $r_0(v) \leftarrow 1/|V|$
7: end for
8: for each $u \in V$ and $v \in V$ do
9:    if $u \in in(v)$ then
10:      $e(u, v) \leftarrow h(u, v)/|out(u)|$
11:   end if
12:   if $v \in out(u)$ then
13:      $\bar{e}(u, v) \leftarrow (1 - h(u, v))/|out(u)|$
14:   else
15:      $\bar{e}(u, v) \leftarrow 1$ {$u$ is a terminal node}
16:   end if
17:   if $v \notin I$ then
18:      $w(u, v) \leftarrow e(u, v)$
19:   else
20:      $w(u, v) \leftarrow e(u, v) + \bar{e}(u, v)/|I|$
21:   end if
22:   $a(u, v) \leftarrow a(u, v) + w(u, v)$
23: end for
24: repeat
25:   for each $v \in V$ do
26:      for each $v \in V$ do
27:         $r_{t+1}(v) \leftarrow a(u, v) \times r_t(u)$
28:      end for
29:   end for
30:   $t \leftarrow t + 1$
31: until $r_{t+1}(v) = r_t(v), \forall v \in V$
32: return $r_t^v, \forall v \in V$
33: end procedure
```

## 5 Illustration

This section illustrates the proposed approach in details. Consider a simple but realistic network as shown in Figure 3, where there are two service hosts: *IP1* and *IP2*, and an attacker's workstation, *Attacker*, connecting to each of the servers via a central router. The network has a security requirement that "*no one can obtain a root privilege access to host IP2*".

Three types of vulnerabilities detected by a scanner (e.g., Nessus [3]): (1) CVE-2006-5794 (vulnerability in the sshd Privilege Separation Monitor in OpenSSH Version before 4.5) (2) CVE-2006-5051 (a signal handler race condition in OpenSSH Version before 4.4), and (3) CVE-2004-0148 (a configuration problem on the *restricted-gid* option). The first two can be exploited remotely to bypass the authentication process (thus, maintain a *user* access level in a victim host), and to obtain a denial of service (thus, gain a *root* access level in the victim host), respectively. Local users can exploit the last vulnerability to bypass access restrictions by changing their access permissions of a home directory via the *ftp*, which causes



a) Mehta et al.'s approach
b) Proposed approach

Figure 2: Comparing assumptions at a terminal node

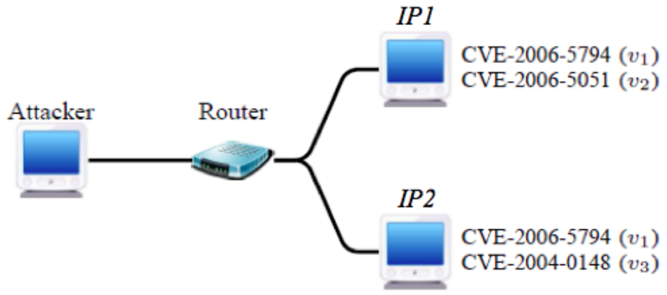
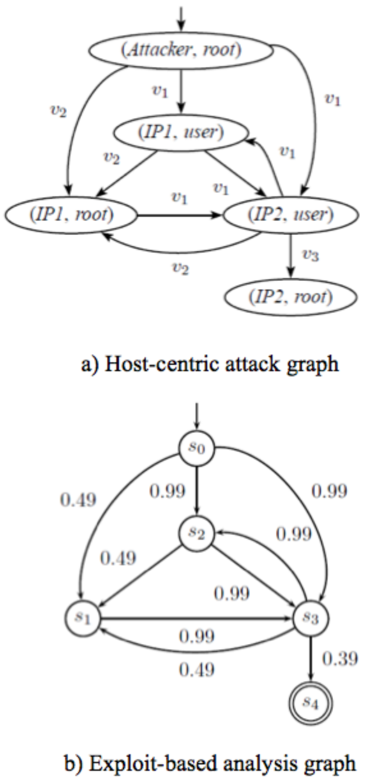


Figure 3: A simple scenario



a) Host-centric attack graph

b) Exploit-based analysis graph

Figure 4: Annotated attack model for analysis

Table 2: Vulnerability and exploitability

| Vulnerability Exploit | Vulnerability | Exploitability |
|---|---|---|
| $v_1$ | CVE-2006-5794 | 0.99 |
| $v_2$ | CVE-2006-5051 | 0.49 |
| $v_3$ | CVE-2004-0148 | 0.39 |

its service program, *wu-ftpd* to, instead, allow access of the root directory. We annotate each configuration of the network in Figure 3 with its corresponding vulnerabilities and their associated labels. For example, *IP2* has two vulnerabilities, namely CVE-2006-5794 (or $v_1$) and CVE-2004-0148 (or $v_3$). More details of these common standard vulnerabilities are described in [7, 20]. Although our approach can be applied to any form of a security model, in this study we use a host-centric attack graph model [8]. Suppose the goal of an attacker is to violate a security requirement. Based on the network configurations and the vulnerabilities shown in Figure 4, we can automatically generate a host-centric attack model as shown in Figure 4a) by employing a model-checking tool such as NuSMV [6] as illustrated in [8].

Each state is labeled by a tuple representing a host name and its access level obtained by an attacker. Thus, (*Attacker*, *root*) is an initial state since an attacker has a root access privilege on his own machine. The attacker's goal is to obtain a root access to *IP2* and thus, (*IP2*, *root*) represents a goal state. In Figure 4b), we rename the states (*Attacker*, *root*), (*IP1*, *root*), (*IP1*, *user*), (*IP2*, *user*) and (*IP2*, *root*) as $s_0$, $s_1$, $s_2$, $s_3$, and $s_4$, respectively.

Table 2 shows the exploitability computed for each of the relevant vulnerabilities obtained from publically known CVSS as described in previous section. Based on the heuristic values in Table 2, we obtained the corresponding attack graph for analysis as shown in Figure 4b) by replacing a state transition of each vulnerability exploit by a corresponding exploitability from Table 2.

The model obtained in Figure 4b) is used for computing a stochastic matrix $A = (a_{ij})$ in the *ExploitRank* algorithm to estimate a probability of transitions between any two attack states. The normalization is required so that the sum of the probabilities of all possible transitions from each state would be one. Note that for each applicable exploit of exploitability $p$, an attacker has two possible transitions: pursuing the exploit to the next state with likelihood $p$, or *not* pursuing the exploit and moving to an initial state to start over with probability (1$p$). Therefore, the sum of probabilities of all possible transitions for *each exploit* is one. Thus, to obtain the normalized stochastic matrix, each probability of exploit from state $s$ to state $t$ is normalized by a total number of applicable exploits from $s$. For example, in Figure 4b), there are three applicable exploits to advance from state $s_0$, to states $s_1$, $s_2$ and $s_3$ with exploitability values 0.49, 0.99, and 0.99, respectively. Thus, the probability of applying the exploit to make a transition from $s_0$ to $s_1$ can be estimated from the normalized heuristic value of $0.49/3 = 0.16$. Similarly, the transition probabilities from $s_0$ to $s_2$ and $s_3$ can be estimated to 0.33 and 0.33, respectively. By the above argument, it is clear that the probability of a transition from $s_0$ (an initial state) to $s_0$ can be estimated from the normalized sum of probabilities of *not* pursuing all the three exploits from $s_0$ yielding a likelihood of entering $s_0$ to be $(0.51 + 0.01 + 0.01)/3 = 0.18$. These results are shown in the first row of the matrix below.

$$\begin{array}{c c} & \begin{array}{ccccc} s_0 & s_1 & s_2 & s_3 & s_4 \end{array} \\ \begin{array}{c} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} & \begin{pmatrix} 0.18 & 0.16 & 0.33 & 0.33 & 0 \\ 0.01 & 0 & 0 & 0.99 & 0 \\ 0.26 & 0.245 & 0 & 0.495 & 0 \\ 0.38 & 0.16 & 0.33 & 0 & 0.13 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Applying the above transition matrix to the *ExploitRank* algorithm, Table 3 shows the results of the intrusion probability distribution obtained by iteration. As shown in Table 3, a steady state is reached in iteration 16, where we obtain the intrusion likelihoods of each state.

Table 3: Computing intrusion likelihoods

| $t$ | $r_t(s_0)$ | $r_t(s_1)$ | $r_t(s_2)$ | $r_t(s_3)$ | $r_t(s_4)$ |
|---|---|---|---|---|---|
| 0 | 0.200 | 0.200 | 0.200 | 0.200 | 0.200 |
| 1 | 0.366 | 0.113 | 0.132 | 0.363 | 0.026 |
| 2 | 0.265 | 0.148 | 0.240 | 0.297 | 0.047 |
| ... | ... | ... | ... | ... | ... |
| 15 | 0.274 | 0.147 | 0.200 | 0.335 | 0.043 |
| 16 | 0.274 | 0.147 | 0.200 | 0.335 | 0.043 |

The results of ranking attack states in the host-centric attack model are shown in the first column of Table 4. We then apply Mehta et al.'s ranking approach that does not employ the exploitability heuristic and obtain the results in the second column of Table 4.

Table 4: Comparisons of ranking results

| State | Our approach | Mehta et al.'s approach |
|---|---|---|
| $s_0$ | 0.274 | 0.150 |
| $s_1$ | 0.147 | 0.145 |
| $s_2$ | 0.200 | 0.102 |
| $s_3$ | 0.335 | 0.209 |
| $s_4$ | 0.043 | 0.394 |

As shown in Table 4, using exploitability heuristics (i.e., our approach) gives a ranking result of $\langle s_3, s_0, s_2, s_1, s_4\rangle$, whereas not using any heuristics (i.e., Mehta et al.'s approach) gives a ranking result of $\langle s_4, s_3, s_0, s_1, s_2\rangle$. Mehta el al.'s approach and ours suggest that $s_4$ and $s_3$, respectively has the highest (relative) likelihood of being attacked (i.e., most vulnerable). However, based on the structure of the attack model in Figure 4b), every path from $s_0$ to $s_4$ must pass thru $s_3$. Therefore, attacking $s_4$ is harder than $s_3$. Thus, the intrusion likelihood of $s_3$ should be higher than that of $s_4$. This is consistent with our ranking result but not Mehta et al.'s.

To further compare the two ranking results, both agree that the initial state $s_0$ is more vulnerable than $s_1$, and $s_2$ since the attacker has already intruded the initial state. However, the ranking order between $s_1$ and $s_2$ are in conflict. Consider an attack from the initial state. As shown in Figure 4b), to reach state $s_1$ (e.g., from $s_0$, $s_2$ or $s_3$) requires exploiting vulnerability $v_2$, whereas to reach state $s_2$ (e.g., from $s_0$ or $s_3$) requires exploiting vulnerability $v_1$. However, according to the CVSS standard, since $exploitability(v_1) = 0.99$ but $exploitability(v_2) = 0.49$, $v_1$ is more vulnerable than $v_2$. Therefore, intruding $s_2$ (via $v_1$) is easier than $s_1$ (via $v_2$). For example, from initial state $s_0$, reaching $s_2$ requires $v_1$ exploit compared to a $v_2$ exploit or a chain of $v_1$ and $v_1$ exploits to reach $s_1$. Therefore, $s_2$ should rank higher than $s_1$. This intuitive reasoning conforms to our ranking order but contradicts with the ranking order produced by Mehta et al.'s approach. In this particular example, using exploitability heuristic based on vulnerability appears to offer more sensible ranking results that obtained without the use of heuristic knowledge.

# 6 Experiments

This section describes two sets of experiments to assess the performance of our approach on relatively large attack models. The first aims to evaluate ranking results of relatively large attack model and the second focuses on computational cost for large-scale models.

## 6.1 Ranking Large Models

Figure 5 shows an attack graph of 66 nodes studied in [23]. The graph was generated with a security property that the intruder would never attain root privileges on the *Linux* host. As described in [23], Figure 5 shows the shaded nodes to signify areas that the intrusion detection system (IDS) alarm has been sounded. Thus, it is possible for the intruder to escape the detection by attacking a portion to the right of the graph that is not "covered" by the IDS. Figure 5 highlights an example of such an attack scenario (a path with solid square nodes), where each attack step identified by exploit number and name. Here the goal states, shown by double circled nodes, are when the intruder violates the security property (i.e., he successfully gains a root privilege on the *Linux* host). Here the attack model has 16 goal states and one single initial state.

It is clear that making a decision on which vulnerability and security flaws to fix first in order to effectively protect the network can be a complex task, especially when dealing with a large attack model. In this context, we ran the ExploitRank algorithm to rank nodes in the attack graph in order of their intrusion likelihoods. To evaluate the ranking results, since there is no known solution, we compare our results with those obtained from Metha

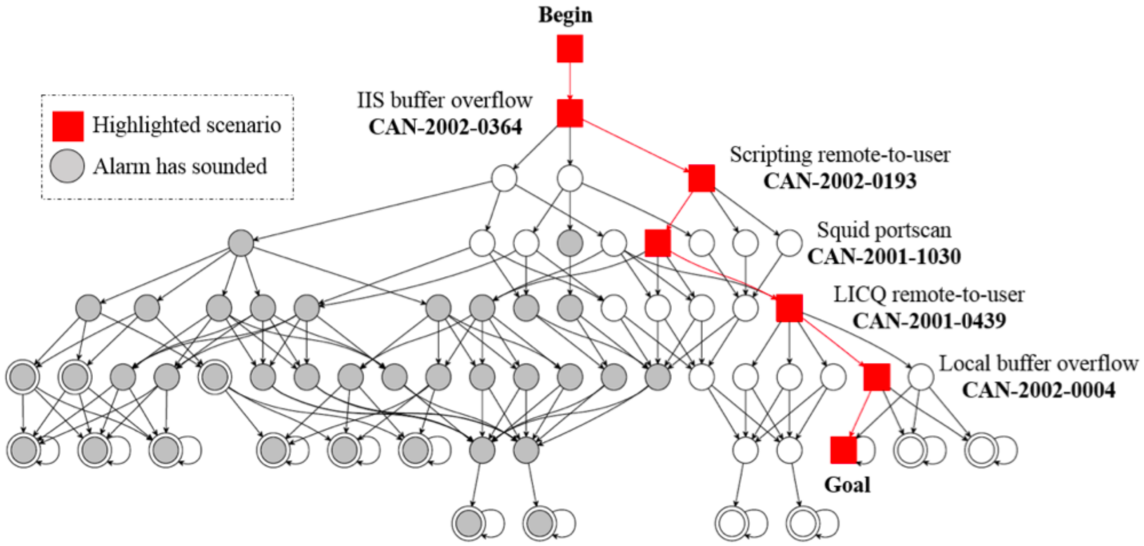


Figure 5: Attack model does not have a full coverage from an IDS [23]

et al.'s approach. Using the attack model, excluding the root, of Figure 5, Figures 6 and 7 show results obtained by Mehta et al.'s approach and ours, respectively.

Recall that Mehta et al.'s approach assumes that every child node is equally likely to be attacked from the parent node, whereas ours differentiates each possibility based on the exploitability of a corresponding vulnerability to be exploited. In this experiment, we assign three exploitability values: 0.1, 0.3, and 0.7 as shown by a dash, solid, and thick solid line to represent the exploit that is hard, somewhat hard, and easy to perform, respectively.

Nodes with equal resulting likelihoods (ranking scores) are labeled with the same rank. As shown in Figure 6, ranking results of Mehta et al.'s approach are the same for nodes that are on the same level with the same degrees of exposure (i.e., incoming arrows), e.g., ranks 6, 14, 43, 53, 61, and so on. The reason for the former (nodes of the same level are of the same rank) is because of the use of Markov property where intrusion likelihoods of states in a current level are impacted by only intrusion likelihoods of states in a previous level, whereas the latter (nodes of the same degree of exposure) is by the assumption on equal likelihoods of attacks from a parent to every child. However, in practice, this is highly unlikely the case, as we know that intrusion likelihoods depend on types of exploits and their exploitability degrees.

Figure 7 shows the ranking results obtained by our approach using a random exploitability assignment as described earlier. Glancing at the results, we no longer obtain the same regularity as observed by Mehta et al.'s approach. It is not necessary that nodes of the same level and the same parent would have the same rank. Both approaches give the root to have the highest rank since it is the easiest to intrude (since the intruder is already there). However, Mehta et al.'s results rank the four goal states at the bottom to be next easiest to intrude. This could be due to the persistence (damping) factor and the assumption that attackers who intrude nodes with no outgoing link (terminal nodes) will persist on their attempt to attack with a probability of the damping factor before giving up to start over (as shown in Figure 2). To compare results of the two approaches in more details, consider three representative scenarios as summarized in Table 5. The number entries, as marked in Figures 6 and 7, represent ranking labels of the nodes, from left to right, in the corresponding depth level of each scenario.

Table 5: Comparison of three ranking scenarios

| Scenario | Mehta et al.'s approach | ExploitRank |
|---|---|---|
| Level 1 | 6, 6, 6 | 2, 1, 2 |
| Level 4 | 53, 53 | 44, 56 |
| Bottom level | 4, 4 | 26, 32 |

Scenario 1 compares three nodes in Level 1 from the root. Mehta et al.'s approach reports that the three nodes are of the same rank. This is clearly wrong. In Figure 7, node 1 has higher rank (more likely to be intruded) than the other two nodes 2. This is because:

1) all the three nodes are intruded by exploiting vulnerability from the same node 0 (root), and

2) the second exploit has the highest exploitability degree (i.e., easiest to intrude), therefore its destination node 1 should have the highest rank.

Since the other two have the same exploitability that is somewhat hard to exploit, they both must be of lower rank than node 1, hence nodes 2.

Scenario 2 compares relative ranks of two nodes in Level 4. In Figure 7, consider nodes 44 and 56. Both have two incoming exploits from the same parent nodes

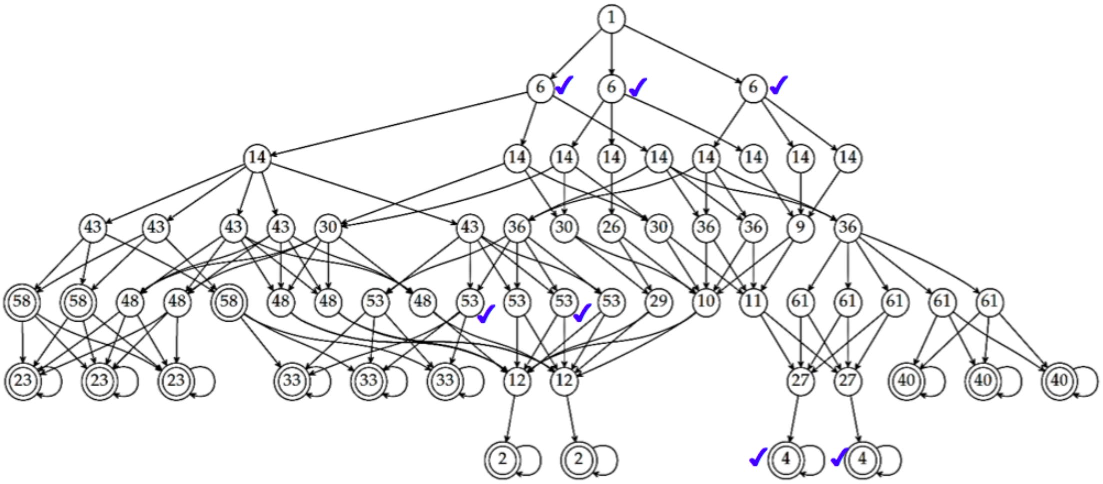


Figure 6: Ranking results by Mehta et al.'s approach

at Level 3. However, both of the incoming exploits to node 44 have exploitability of 0.3 (somewhat hard to exploit), while both of the incoming exploits to node 56 are of exploitability 0.1 (hard to exploit). Therefore, node 44 has a higher order of intrusion likelihood than node 56 as ranked by our approach. However, Mehta et al.'s approach results in equal rank for nodes 53.

Finally, Scenario 3 compares relative ranks of two terminal nodes at the bottom level of the tree. In Figure 7, consider nodes 26 and 32. Since each can only be intruded by exploiting from its parent who has the same intrusion likelihood (i.e., the same rank of 22), the destination of a dark solid link (high exploitability of 0.7), node 26 must be easier to intrude. Thus, node 26 has a higher intrusion likelihood than node 32 as obtained by our approach. Once again, Mehta et al.'s approach does not distinguish such likelihoods. Based on the three case scenarios, *ExploitRank* algorithm outperforms Mehta et al.'s approach. Although we do not compare all possible relative ranking results, we anticipate that our approach would rank correctly based on the logics of our recurrence formulae.

In addition, we have also experimented with a modified Metha et al.'s approach where exploitability is used as a heuristic for estimating prior probability for an exploit. The ranking results of our approach still outperform those of the modified Mehta et al.'s approach (not shown here). For example, consider a relative ranking of 26 and 32 in Figure 7. The modified Metha et al.'s approach that uses exploitability as heuristic gives the same ranking score of 3 for these two nodes. This is clearly wrong since both nodes have parents of the same ranking order; each can be reached by a single exploit where one has a higher exploitability degree than the other. Therefore, the resulting ranks of these two nodes should be different. The main distinction that contributes to this significant difference is due to the fact that Mehta et al.'s approach assumes that an attacker behaves like a surfer when he reaches a terminal node in that there is a chance that we would continue penetrating the node intrusion (or surfing the site with the likelihood of a damping factor), while our approach does not. As a result, Mehta et al.'s assumption increases the intrusion likelihoods of terminal nodes and lessens the impact of the degree of exploitability of the exploits to reach these nodes. Unlike Mehta et al.'s approach, we assume that the attacker starts over when he reaches the terminal point of the attack path.

## 6.2 Performance on Large-scale Attack Models

This section presents experiments to see if the proposed approach can be computed efficiently enough to cope with large-scale attack models. Our *ExploitRank* algorithm was implemented using NodeJS language on Ubuntu Linux machine with an Intel Core i5 CPU of 3.20 GHz and 2 GB memory.

Table 6 shows a sample of running times of the implemented algorithm with various sizes (number of nodes and edges) of attack graphs. The edges were randomly generated. As shown in Table 6, while the size of the graph roughly grows with a constant rate of four, the running times grow approximately at the rates of 4, 5, 2 and 3.5, respectively, making the ratios between the size growth and the running time growth close to one (except for the case of 1024 nodes).

Table 6: Running times of our ranking approach

| #Nodes | #Edges | Graph Size | Running time (sec) |
|---:|---:|---:|---:|
| 128 | 8,192 | 8,320 | 0.5 |
| 256 | 32,768 | 33,024 | 2.1 |
| 512 | 131,072 | 131,584 | 10.7 |
| 1,024 | 524,288 | 525,312 | 20.2 |
| 2,048 | 2,097,152 | 2,099,200 | 71.5 |

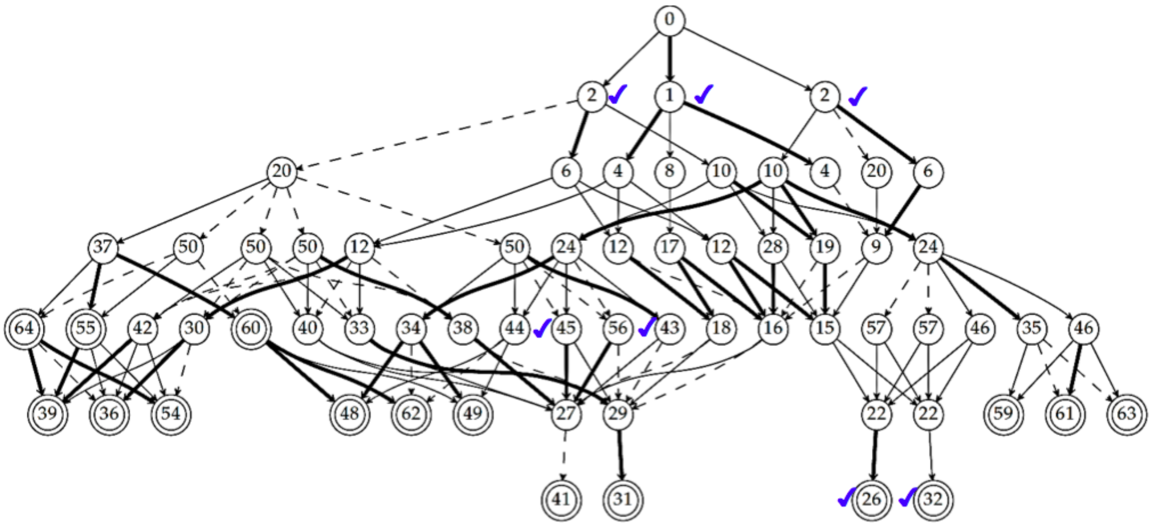


Figure 7: Ranking results by *ExploitRank* algorithm

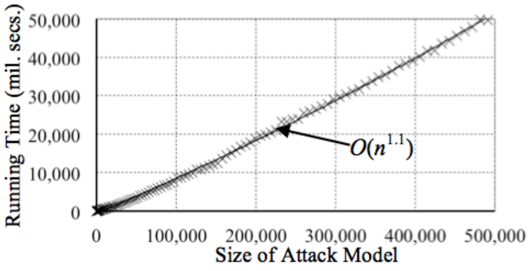


Figure 8: Performance of the ranking algorithm

To further evaluate the performance of our approach, we ran 500 runs of experiments with various sizes of attack models ranging from 60 to 501,000. Figure 8 shows the resulting runtimes in milliseconds. The running time obtained fits to an approximate linear equation: $0.0185n^{1.13}$, where $n$ is the attack graph size. Just like PageRank algorithm that can handle ranking of a huge number of web pages, based on similar concepts of Markov Model, our ExploitRank algorithm can scale with linear time in size of the graph. The advantage of ranking algorithm is that one can give the number of top $k$ nodes to be ranked. This is useful when resources are limited.

## 7 Conclusions

We present an automated approach to attack model analysis that allows quantitative ranking of network nodes by their intrusion likelihoods. What sets our approach apart from the rest is our use of domain-specific knowledge that can be obtained from public databases or derived in a principled way from the structure of the network. The approach is adapted from a Markov Model-based ranking algorithm that is well-established tractable computational model used for intractable problems (e.g., ranking webpages).

This paper differs from our previous work [11] in that the previous work extends Mehta et al.'s approach to using the exploitability concept. However, as we have illustrated in this paper in the example in Section 6.1 that the basic assumption adapted by Mehta et al.'s approach is not appropriate for use in intrusion analysis. Future work includes additional evaluations of the proposed approach by investigating a large network in real-world applications.

## References

[1] P. Ammann, J. Pamula, J. Street, and R. Ritchey, "A host-based approach to network attack chaining analysis," in *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC'05)*, pp. 72-84, 2005.

[2] S. R. Asumssen, "Markov Chains", *Applied Probability and Queues: Stochastic Modelling and Applied Probability*, vol.51, pp. 3-38, 2003.

[3] J. Beale, R. Deraison, H. Meer, R. Temmingh, and C. V. D. Walt, *Nessus Network Auditing*, Syngress Publishing, 2004.

[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107-117, 1998.

[5] CIAC, "Computer incident advisory capability (CIAC)," 2009. (http://www.ciac.org/ciac/index.html)

[6] A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella, "Nusmv 2: An opensource tool for symbolic model checking," in *Proceedings of the 14th*

*International Conference on Computer Aided Verification (CAV'02)*, pp. 359-364, 2002.

[7] CVSS, "Common vulnerability scoring system (cvss)," 2009. (http://www.first.org/cvss/cvss-guide.html)

[8] R. Hewett and P. Kijsanayothin, "Host-centric model checking for network vulnerability analysis," in *Proceedings of the 2008 Annual Computer Security Applications Conference (ACSAC'08)*, pp. 225-234, 2008.

[9] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Computer Security Applications Conference*, pp. 121-130, 2006.

[10] S. Jha, O. Sheyner, and J. M. Wing, "Two formal analys s of attack graphs," in *Proceedings of the 15th IEEE workshop on Computer Security Foundations (CSFW'02)*, PP. 49, 20029.

[11] P. Kijsanayothin and R. Hewett, "Exploit-based analysis of attack models", in *Proceeding of the 12th International Symposium on Network Computing and Applications (NCA'13)*, pp. 183-186, 2013.

[12] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604-632, Sep. 1999.

[13] K. Lye and J. M. Wing, "Game strategies in network security," *International Journal of Information Security*, vol. 4, no. 1-2, pp. 71-86, 2005.

[14] V. Mehta, C. Bartzis, H. Zhu, E. M. Clarke, and J. M. Wing, "Ranking attack graphs," in *Recent Advances in Intrusion Detection*, LNCS 4219, pp. 127-144, 2006.

[15] NIST, "The united state national institute of standard and technology," 2009. (http://www.nist.gov/index.html)

[16] S. Noel and S. Jajodia, "Managing attack graph complexity through visual hierarchical aggregation," in *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC'04)*, pp. 109-118, 2004.

[17] S. Noel and S. Jajodia, "Understanding complex network attack graphs through clustered adjacency matrices," in *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC'05)*, pp. 160-169, 2005.

[18] NVD, "National vulnerability database (NVD)," National Institute of Science and Technology (NIST), 2009. (http://nvd.nist.gov/nvd.cfm)

[19] R. E. Sawilla and X. Ou, "Identifying critical attack assets in dependency attack graphs," in *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS'08)*, pp. 18-34, 2008.

[20] K. Scarfone and P. Mell, "An analysis of cvss Version 2 vulnerability scoring," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09)*, Washington, pp. 516-525, 2009.

[21] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy (SP'02)*, pp. 273, 2002.

[22] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," *DARP A Information Survivability Conference and Exposition*, vol. 2, pp. 1307, 2001.

[23] J. M. Wing, "Scenario graphs applied to security", *Verification of Infinite-State Systems with Applications to Security,* IOS Press, pp. 229-233, 2005.

**Rattikorn Hewett** is a professor and Chair of the department of Computer Science, Texas Tech University. She has a Ph.d. in Computer Science from Iowa State University, an M. Eng. Sc. in Computer Science from the University of New South Wales, and a B.A. (Hons) in Pure Mathematics and Statistics from Flinders University, Australia. She was a post-doctoral fellow at the Knowledge System Laboratory, Stanford University. Her research in Artificial Intelligence (AI) includes intelligent control, and machine learning with applications in data mining, bioinformatics and software engineering. Her recent work involves applied AI research to automated science, network security and Internet security. She has published extensively and has served on numerous international conference and work-shop program committees.

**Phongphun Kijsanayothin** received his B. Eng. (Computer Engineering) from the King Monkuts Institute of Technology Ladkrabang, in 1999, and his M.Eng. (Computer Engineering) from Kasetsart University, in 2003, and his Ph.D. in Computer Science from Texas Tech University, in 2010. Currently, he is an assistant professor in the Department of Electrical and Computer Engineering, Naresuan University. His research interests include network security, cyber security and software testing in the context of object-oriented development.

# A Novel Untraceable Authentication Scheme for Mobile Roaming in GLOMONET

Hai-Duong Le[1], Chin-Chen Chang[1,2], and Yeh-Chieh Chou[1]
*(Corresponding author: Chin-Chen Chang)*

Department of Information Engineering and Computer Science, Feng Chia University[1]
Taichung, 40724, Taiwan, R.O.C.
Department of Computer Science and Information Engineering, Asia University[2]
Taichung, 41354, Taiwan, R.O.C.
(Email: alan3c@gmail.com) *(Received Dec. 22, 2014; revised and accepted Feb. 22 & Mar. 6, 2015)*

## Abstract

In global mobile network, it is required to authenticate mobile users, provide secure communication channel between a user and a foreign agent using session key, and guarantee users' anonymity and untraceability. In order to improve the security of mobile roaming service, two-factor authentication which employs smart card and password was introduced to global mobile network. In 2014, Kuo et al. [5] proposed an anonymous two-factor authentication scheme for mobile roaming service. However, we found that this scheme is vulnerable to four kinds of man-in-the-middle attacks and denial-of-service attack. In this paper, we first review Kuo et al.'s scheme and analyze its weaknesses. Then, we propose an efficient anonymous two-factor authentication protocol that overcomes those vulnerabilities in Kuo et al.'s.

*Keywords: Anonymity, authentication, GLOMONET, mobile roaming, untraceability*

## 1 Introduction

Mobile telecommunications technology has developed at a rapid pace. The 3G and 4G networks have been deployed all over the world providing mobile users with broadband Internet access. When a user travels from one place to another, the continuity of the mobile service is enforced by the mobile roaming service, which is also known as the global mobility network (GLOMONET) [8, 11]. Along with the great advantages that the mobile networks provide, there are security challenges that we need to overcome. Because of its nature, data transmission in mobile network is susceptible to eavesdropping and interception. Therefore, establishing a secure channel between a user's device and a foreign agent is a must in GLOMONET. Moreover, the personal information of a user, such as his/her identity, location, travelling, Internet accessing habits, etc., should be kept confidential. Thus, both anonymity and untraceability are the characteristics that the global mobile networks must ensure.

In 2004, Zhu and Ma [13] first proposed an anonymous authentication scheme based on hash function for global mobile network. In 2006, Lee et al. [6] pointed out that Zhu-Ma's scheme cannot achieve mutual authentication and perfect backward secrecy, and it is vulnerable to forgery attack. Lee et al. then proposed a scheme to overcome these weaknesses. However, Wu et al. [9] and Chang et al. [2] showed that both Lee et al.'s and Zhu-Ma's schemes failed to ensure anonymity. Later, Youn at al. [12] demonstrated that Chang et al.'s protocol does not provide secure key establishment and anonymous authentication. In 2012, Mun et al. [7] illustrated that Wu et al.'s scheme cannot achieve anonymity and perfect forward secrecy. Recently, Kim and Kwak [4] showed that Mun et al.'s scheme is susceptible to replay attacks and man-in-the-middle attacks.

In 2013, Xie et al. [10] proposed a new anonymous two-factor authentication scheme for roaming service in GLOMONET. In this scheme, a mobile user must possess a smart card and memorize a password in order to authenticate with the mobile network agents. The computation cost of Xie et al.'s scheme is high due to employing both modular exponentiation and symmetric encryption/decryption operations. Furthermore, He et al. [3] found that Xie et al.'s protocol fails to prevent two types of impersonation attack; and they proposed another scheme that resolves these weaknesses.

In 2014, in line with Xie et al.'s two-factor authentication scheme, Kuo et al. [5] proposed an efficient anonymous authentication protocol for mobile roaming service. Kuo et al.'s protocol is more efficient in computing than Xie et al.'s. However, we found that Kuo et al.'s scheme is prone to four kinds of man-in-the-middle attacks and denial-of-service attack. In this paper, we first demonstrate how an attacker can exploit the weaknesses in Kuo et al.'s. Then, we propose a novel and secure anonymous two-factor authentication scheme for global mobile network.

The rest of the paper is organized as follows. Section 2 reviews Kuo et al.'s scheme and illustrates its weaknesses. Our proposed protocol is introduced in Section 3. The security analysis is provided in Section 4. After that, we evaluate the security and performance of our scheme in Section 5. Finally, we conclude the paper in Section 6.

# 2 Related Work

In this section, we briefly review Kuo et al.'s scheme and demonstrate its weaknesses. At first, we list all the notations used in this paper in Table 1.

Table 1: Notations

| | |
|---|---|
| $MU$ | The mobile user |
| $ID_{MU}$ | The identity of $MU$ |
| $FA$ | The foreign agent |
| $ID_{FA}$ | The identity of $FA$ |
| $HA$ | The home agent |
| $ID_{HA}$ | The identity of $HA$ |
| $pw_{MU}$ | The password of $MU$ |
| $\mathcal{A}$ | The adversary |
| $SC$ | The smart card |
| $h(\cdot)$ | The hash operation |
| $P$ | A point on the elliptic curve |
| $P.x$ | The value of $P$ on $x$-axis |
| $s$ | $HA$'s long-term secret |
| $r$, $r_1$, $r_2$, $N_{MU}$ | Random numbers |

## 2.1 Review of Kuo et al.'s Scheme

There are four phases in Kuo et al.'s scheme: registration phase, authentication and establishment of the session key phase, update session key phase, and password change phase.

### 2.1.1 Registration Phase

In this phase, $MU$ registers with $HA$ in order to use roaming service. $MU$ and $HA$ execute the following steps:

Step 1. $MU$ chooses an identity $ID_{MU}$ and a password $pw_{MU}$, and then computes $PW_{MU} = h(ID_{MU} \| pw_{MU})$. It sends $\{ID_{MU}, PW_{MU}\}$ to $HA$ via a secure channel.

$$MU \rightarrow HA : m_{reg} = \{ID_{MU}, PW_{MU}\}.$$

Step 2. $HA$ checks whether $ID_{MU}$ is available for use. If it is, $HA$ chooses a random nonce $N_{MU_i}$ and $p_{HA-MU_i}$, and computes $U = h(p_{HA-MU_i} \| N_{MU_i})$, $W_i = PW_{MU} \oplus N_{MU_i}$, and $V_i = N_{MU_i} \oplus p_{HA-MU_i}$. It then writes $\{ID_{HA}, W_i, V_i, h(\cdot)\}$ to a smart card and issues it to $MU$. Finally, $HA$ stores the values $U$, $PW_{MU}$, and $p_{HA-MU_i}$ in its database.

$$\begin{aligned} HA &\rightarrow MU : SC = \{ID_{HA}.W_i, V_i, h(\cdot)\}, \\ HA &\rightarrow DB : \{U, PW_{MU}, p_{HA-MU_i}\}. \end{aligned}$$

### 2.1.2 Authentication and Establishment of the Session Key Phase

In this phase, $HA$ helps $FA$ and $MU$ authenticating each other as follows:

Step 1. $MU$ inserts the smart card into the reader and provides $ID_{MU}$ and $pw_{MU}$. The smart card chooses a random nonce $N_{MU_{i+1}}$, and derives $PW_{MU} = h(ID_{MU} \| p_{MU})$, $N_{MU_i} = PW_{MU} \oplus W_i$, $p_{HA-MU_i} = N_{MU_i} \oplus V_i$. Then, it computes $S_1 = h(p_{HA-MU_i} \| N_{MU_i})$, $S_2 = PW_{MU} \oplus N_{MU_{i+1}}$, $S_3 = h(N_{MU_{i+1}} \| ID_{FA})$, $S_4 = h(PW_{MU} \oplus h(p_{HA-MU_i} \| N_{MU_{i+1}}))$, and sends $m_1 = \{ID_{HA}, S_1, S_2, S_3, S_4\}$ to $FA$ after saving $N_{MU_{i+1}}$.

$$MU \rightarrow FA : m_1 = \{ID_{HA}, S_1, S_2, S_3, S_4\}.$$

Step 2. $FA$ chooses a random nonce $a$ and computes $aP$. It sends $\{ID_{FA}, S_1, S_2, S_3, S_4, aP\}$ to $HA$, and stores $ID_{HA}$, $a$, $aP$.

$$FA \rightarrow HA : m_2 = \{ID_{FA}, S_1, S_2, S_3, S_4, aP\}.$$

Step 3. Upon receiving $m_2$, $HA$ uses $S_1$ to search the database and retrieves $PW_{MU}$, $p_{HA-MU_i}$. It derives $N_{MU_{i+1}} = S_2 \oplus PW_{MU}$, and computes $S_3' = h(N_{MU_{i+1}} \| ID_{FA})$, and $S_4' = h(PW_{MU} \oplus h(p_{HA-MU_i} \| N_{MU_{i+1}}))$. $HA$ checks whether $S_3' \stackrel{?}{=} S_3$ and $S_4' \stackrel{?}{=} S_4$. If they both hold, $HA$ successfully authenticates $MU$, and $FA$; otherwise, it informs $FA$ to terminate the session. Next, it computes $S_5 = h(PW_{MU} \| N_{MU_{i+1}})$, $S_6 = h(ID_{FA} \| ID_{HA} \| S_5)$, and $S_7 = h(aP.x \| S_5)$, and replaces $S_1$ in the database with $h(p_{HA-MU_i} \| N_{MU_{i+1}})$. It then sends $\{ID_{HA}, S_6, S_7\}$ to $FA$.

$$HA \rightarrow FA : m_3 = \{ID_{HA}, S_6, S_7\}.$$

Step 4. $FA$ authenticates $HA$ by checking $ID_{HA}$ in its database. If $ID_{HA}$ exists, $FA$ trusts that $HA$ is legitimate and transmits $\{ID_{FA}, S_6, S_7, aP\}$ to $MU$.

$$FA \rightarrow MU : m_4 = \{ID_{FA}, S_6, S_7, aP\}.$$

Step 5. $MU$ verifies whether $S_6 \stackrel{?}{=} h(ID_{FA} \| ID_{HA} \| S_5)$ and $S_7 \stackrel{?}{=} h(p_{HA-MU_i} \| N_{MU_{i+1}})$. If both equations hold, it chooses a random nonce $b$, and computes $bP$, $K_{MF} = h(abP.x)$, and $C_{MF} = h(K_{MF} \| bP.x)$. The smart card updates $W_i$, $V_i$ with $W_{i+1} = PW_{MU} \oplus N_{MU_{i+1}}$, $V_{i+1} = N_{MU_{i+1}} \oplus p_{HA-MU_i}$, respectively, and stores $aP$. Then, it sends $\{bP, C_{MF}\}$ to $FA$.

$$MU \rightarrow FA : m_5 = \{bP, C_{MF}\}.$$

Step 6. $FA$ computes $K_{MF} = h(abP.x)$, and $C'_{MF} = h(K_{MF} \| bP.x)$. It verifies whether $C_{MF} \overset{?}{=} C'_{MF}$. If they are equal, $FA$ successfully authenticates $MU$, and writes $C_{MF}$, $aP$ into its database.

Eventually, $FA$ and $MU$ mutually authenticate each other and share the session key $K_{MF}$.

### 2.1.3 Update Session Key Phase

To update the session key, $MU$ and $FA$ perform the following steps:

Step 1. $MU$ chooses a new random nonce $b_i$ and sends $b_iP$, $C_{MF_i}$ to $FA$.

$$MU \rightarrow FA : m_6 = \{b_iP, C_{MF}\}.$$

Step 2. $FA$ checks the existence of $C_{MF}$ in the database. If there is a record of it, $FA$ trusts $MU$, and retrieves $a_{i-1}P$ from the record. Next, it selects a new random nonce $a_i$, and computes $K_{MF_{i+1}} = h(a_ib_iP.x)$, $C_{MF_{i+1}} = h(K_{MF_{i+1}} \| b_iP.x)$, and $h_1 = h(C_{MF_{i+1}} \| a_{i-1}P.x)$. It then replaces $C_{MF_i}$ by $C_{MF_{i+1}}$ and $a_{i-1}P$ by $a_iP$ in the database before delivering $\{a_iP, h_1\}$ to $MU$.

$$FA \rightarrow MU : m_7 = \{a_iP, h_1\}.$$

Step 3. $MU$ computes the new session key $K_{MF_{i+1}} = h(a_ib_iP.x)$, and $C'_{MF_{i+1}} = h(K_{MF_{i+1}} \| b_iP.x)$. It then checks whether $h_1 \overset{?}{=} h(C'_{MF_{i+1}} \| a_{i-1}P.x)$. If it holds, $MU$ authenticates $FA$; otherwise, it terminates the session. At last, it replaces $C_{MF_i}$ with $C_{MF_{i+1}}$, $a_{i-1}P$ with $a_iP$ in the smart card's memory.

### 2.1.4 Password Change Phase

In this phase, $MU$ changes its password as follows:

Step 1. $MU$ chooses a new password $pw_{MU_{new}}$, and computes $PW_{MU_{new}} = h(ID_{MU} \| pw_{MU_{new}})$, $U = h(p_{HA-MU_i} \| N_{MU_i})$, $h_1 = PW_{MU} \oplus PW_{MU_{new}}$, and $h_2 = h(PW_{MU_{new}} \| p_{HA-MU_i})$. It then transmits $U$, $h_1$, $h_2$ to $HA$.

$$MU \rightarrow HA : m_8 = \{U, h_1, h_2\}.$$

Step 2. $HA$ uses $U$ to search and retrieves $PW_{MU}$, $p_{HA-MU_i}$ from the database. Next, it computes $PW'_{MU_{new}} = PW_{MU} \oplus h_1$, $h'_2 = h(PW'_{MU_{new}} \| p_{HA-MU_i})$. It checks whether $h'_2 \overset{?}{=} h_2$. If the equation holds, $HA$ replaces $PW_{MU}$ with $PW_{MU_{new}}$. It then computes $h_3 = h(PW_{MU} \| p_{HA-MU_i})$, and sends $h_3$ to $MU$.

$$HA \rightarrow MU : m_9 = \{h_3\}.$$

Step 3. $MU$ verifies whether $h_3 \overset{?}{=} h(PW_{MU} \| p_{HA-MU_i})$. If it holds, $MU$ updates $W_i$ with $PW_{MU_{new}} \oplus N_{MU_i}$.

## 2.2 The Weakness of Kou et al.'s Scheme

### 2.2.1 Man-in-the-middle Attack in Authentication Phase ($MU - FA$)

In the authentication phase, an attacker $\mathcal{A}$ intercepts the messages sending between $MU$ and $FA$. It forwards the message $m_1$ from $MU$ to $FA$. Upon intercepting $m_4$ from $FA$ to $MU$, it generates a random nonce $c$, and computes $K^*_{MF} = h(caP.x)$, $C^*_{MF} = h(K^*_{MF} \| cP.x)$, where $aP$ is in $m_4$. Then, it sends the message $m^*_5 = \{cP, C^*_{MF}\}$ to $FA$.

When receiving $m^*_5$, $FA$ computes $K^*_{MF} = h(caP.x)$, and $C^*_{MF} = h(K_{MF} \| cP.x)$. Since $C^*_{MF}$ equals to $C_{MF}$, $FA$ trusts that it is in communication with a legitimate mobile user and the shared session key is $K_{MF}$. At this stage, $\mathcal{A}$ has successfully deceived $FA$, and it impersonates a valid user to use $FA$'s services.

In this attack, the values $K_{MF}$ and $C_{MF}$ are computed from $cP$ and $aP$, where $aP$ is sent in plaintext. $\mathcal{A}$ exploits the fact that $FA$ does not verify whether $cP$ really comes from $MU$ or not.

### 2.2.2 Man-in-the-middle Attack in Update Session Key Phase ($MU - FA$)

In this attack, $\mathcal{A}$ eavesdrops all the messages between $MU$ and $FA$. When $MU$ sends $m_6 = \{b_iP, C_{MF_i}\}$ to $FA$, the attacker intercepts this message and generates a new random nonce $c$. Then, it transmits $m^*_6 = \{cP, C_{MF}\}$ to $FA$.

Once receiving $m^*_6$, $FA$ only searches for the record of $C_{MF_i}$ in its database, but it does not verify whether $cP$ comes from $MU$ or not. Therefore, $FA$ will accept $\mathcal{A}$ as a legitimate mobile user, if it finds a match for $C_{MF}$. It then computes the session key $K_{MF_{i+1}} = h(ca_iP.x)$. At this point, it has succeeded in forging a legitimate $MU$.

### 2.2.3 Man-in-the-middle Attack in Authentication Phase ($MU - FA, FA - HA$)

Since the communication channel between $FA$ and $HA$ is not secure, the attacker $\mathcal{A}$ can intercept the message $m_2$ from $FA$ to $HA$. It generates a new random nonce $c$ and replaces $aP$ in $m_2$ by $cP$. Then, it forwards the modified $m_2$ to $HA$. Because $HA$ does not verify whether

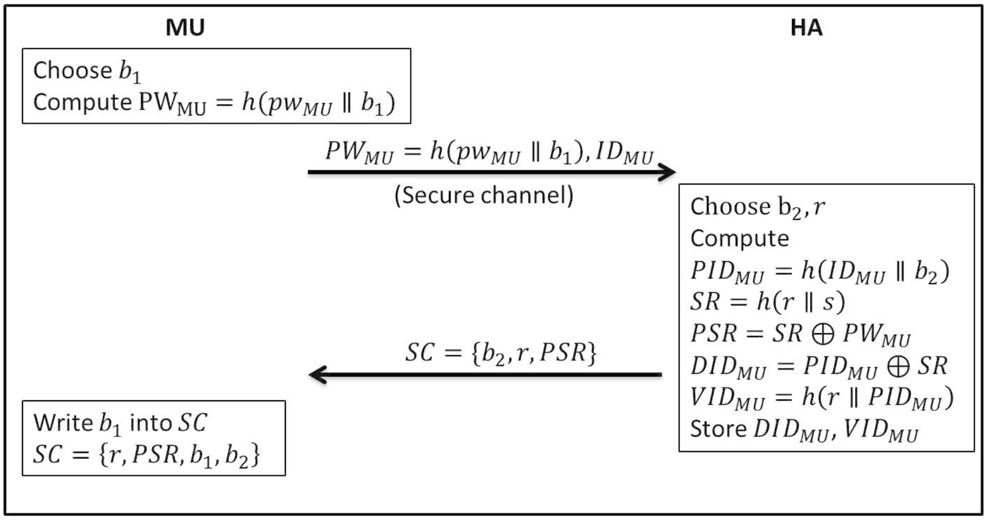

MU
HA
Choose $b_1$
Compute $\mathrm{PW}_{\mathrm{MU}} = h(pw_{MU} \parallel b_1)$
$PW_{MU} = h(pw_{MU} \parallel b_1), ID_{MU}$
(Secure channel)
Choose $\mathrm{b}_2, r$
Compute
$PID_{MU} = h(ID_{MU} \parallel b_2)$
$SR = h(r \parallel s)$
$PSR = SR \oplus PW_{MU}$
$DID_{MU} = PID_{MU} \oplus SR$
$VID_{MU} = h(r \parallel PID_{MU})$
Store $DID_{MU}, VID_{MU}$
$SC = \{b_2, r, PSR\}$
Write $b_1$ into $SC$
$SC = \{r, PSR, b_1, b_2\}$

Figure 1: Registration phase

$cP$ comes from $FA$ or not, it will accept $MU$ and compute the value $S_7 = h(cP.x \| S_5)$ based on $cP$. That means $HA$ has accidentally authenticated $cP$. $\mathcal{A}$ forwards $m_3$ from $HA$ to $FA$. Then, it intercepts $m_4$ from $FA$ to $MU$, and replaces $aP$ with $cP$.

Upon receiving the modified $m_4$, $MU$ verifies $S_6$ and $S_7$; it trusts that $FA$ is valid, and $cP$ comes from $FA$. Then, $MU$ selects a random nonce $b$ and sends $bP$ in $m_5$. Now, the attacker intercepts $m_5$, and computes the session key $K_{MF} = h(caP.x)$. Finally, $\mathcal{A}$ has successfully masqueraded as $FA$.

#### 2.2.4 Man-in-the-middle Attack in Authentication Phase using Stolen Verifiers

Suppose that the attacker can either steal $HA$'s database or access it. Using $U = h(p_{HA-MU_i} \| N_{MU_i})$, $PW_{MU}$, and $pw_{HA-MU_i}$, $\mathcal{A}$ can derives $N_{MU_i} = S_2 \oplus PW_{MU}$, and computes $S_5 = h(PW_{MU} \| N_{MU_{i+1}})$, $S_6 = h(ID_{FA} \| ID_{HA} \| S_5)$, and $S_7 = h(aP.x \| S_5)$, where $S_2$ is in $m_1$ sent from $MU$, and $a$ is generated by $\mathcal{A}$. The attacker then can send $m_4 = \{ID_{FA}, S_6, S_7, aP\}$ to $MU$.

Upon receiving $m_4$ from $\mathcal{A}$, $MU$ verifies $S_6$ and $S_7$, and trusts that it is communicating with a legitimate $FA$. Then, it sends $bP$ to $\mathcal{A}$. After this, the attacker computes the session key $K_{MF} = h(abP.x)$. In the end, $\mathcal{A}$ has successfully deceived $MU$ into thinking of it as a valid $FA$.

#### 2.2.5 Denial-of-Service Attack

$HA$ might face unsynchronization problem when it updates the database without knowing whether $MU$ has completed the authentication phase or not. If $MU$ terminates the session before updating $W_i$ and $V_i$ in the smart card, then it will not be able to authenticate with $HA$ next time.

This may lead to a bigger problem in which an attacker $\mathcal{A}$ can mount DoS attack (Denial-of-Service) to any mobile user. $\mathcal{A}$ can seize any message $m_3$ or $m_4$, and cause the corresponding $MU$ unable to authenticate with $FA$ in the future unless re-registering with $HA$.

## 3 The Proposed Scheme

The proposed scheme shows how a foreign agent ($FA$) authenticates a mobile user ($MU$) with the help of $MU$'s home agent ($HA$). As a result, $FA$ and $MU$ will be mutually authenticated and share a session key. The scheme consists of four phases: registration phase, authentication phase, session key update phase, and password changing phase.

***Assumption.*** In this scheme, we assume that $FA$ and $HA$ are mutually authenticated and they communicate via a secure channel.

### 3.1 Registration Phase

In a mobile network, it is required that a mobile user $MU$ registers with its home agent $HA$. The registration procedure is shown in Figure 1 and has the following steps:

Step 1. First, the mobile user $ID_{MU}$ chooses a password $pw_{MU}$ and a random number $b_1 \in \mathbb{Z}_p^*$. It computes the has $PW_{MU} = h(pw_{MU} \| b_1)$. Then, it submits the registration request to the home agent $HA$ via a secure channel.

$$MU \rightarrow HA : m_{reg} = \{ID_{MU}, PW_{MU}\}.$$

Step 2. Upon receiving the request, the home agent identifies $MU$ and verifies the identity $ID_{MU}$. $HA$ chooses two random numbers $b_2$, $r \in \mathbb{Z}_p^*$

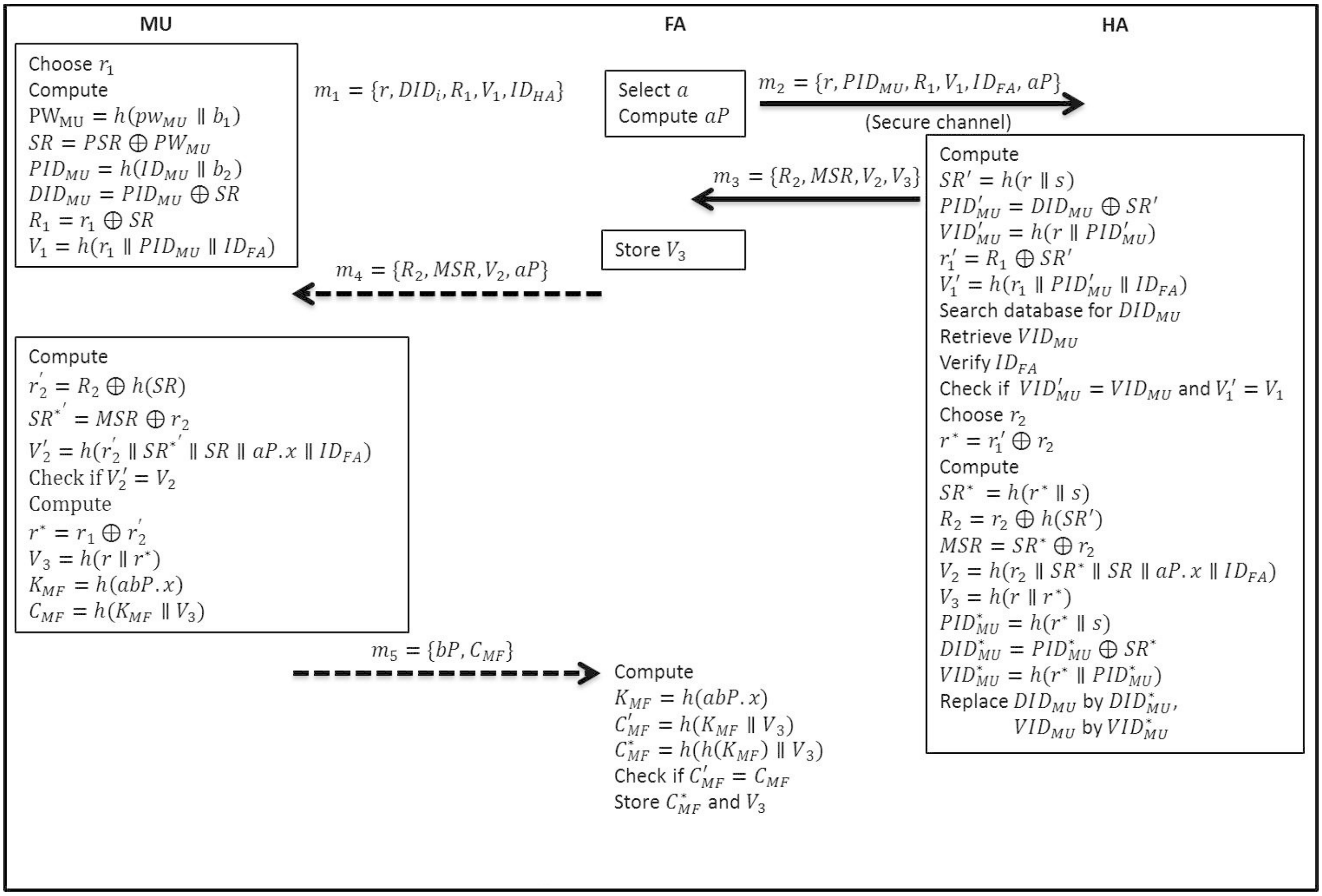

MU
FA
HA
Choose $r_1$
Compute
$PW_{MU} = h(pw_{MU} \| b_1)$
$SR = PSR \oplus PW_{MU}$
$PID_{MU} = h(ID_{MU} \| b_2)$
$DID_{MU} = PID_{MU} \oplus SR$
$R_1 = r_1 \oplus SR$
$V_1 = h(r_1 \| PID_{MU} \| ID_{FA})$
$m_1 = \{r, DID_i, R_1, V_1, ID_{HA}\}$
Select $a$
Compute $aP$
$m_2 = \{r, PID_{MU}, R_1, V_1, ID_{FA}, aP\}$
(Secure channel)
Compute
$SR' = h(r \| s)$
$PID'_{MU} = DID_{MU} \oplus SR'$
$VID'_{MU} = h(r \| PID'_{MU})$
$r'_1 = R_1 \oplus SR'$
$V'_1 = h(r_1 \| PID'_{MU} \| ID_{FA})$
Search database for $DID_{MU}$
Retrieve $VID_{MU}$
Verify $ID_{FA}$
Check if $VID'_{MU} = VID_{MU}$ and $V'_1 = V_1$
Choose $r_2$
$r^* = r'_1 \oplus r_2$
Compute
$SR^* = h(r^* \| s)$
$R_2 = r_2 \oplus h(SR')$
$MSR = SR^* \oplus r_2$
$V_2 = h(r_2 \| SR^* \| SR \| aP.x \| ID_{FA})$
$V_3 = h(r \| r^*)$
$PID^*_{MU} = h(r^* \| s)$
$DID^*_{MU} = PID^*_{MU} \oplus SR^*$
$VID^*_{MU} = h(r^* \| PID^*_{MU})$
Replace $DID_{MU}$ by $DID^*_{MU}$,
$VID_{MU}$ by $VID^*_{MU}$
$m_3 = \{R_2, MSR, V_2, V_3\}$
Store $V_3$
$m_4 = \{R_2, MSR, V_2, aP\}$
Compute
$r'_2 = R_2 \oplus h(SR)$
$SR^{*'} = MSR \oplus r_2$
$V'_2 = h(r'_2 \| SR^{*'} \| SR \| aP.x \| ID_{FA})$
Check if $V'_2 = V_2$
Compute
$r^* = r_1 \oplus r'_2$
$V_3 = h(r \| r^*)$
$K_{MF} = h(abP.x)$
$C_{MF} = h(K_{MF} \| V_3)$
$m_5 = \{bP, C_{MF}\}$
Compute
$K_{MF} = h(abP.x)$
$C'_{MF} = h(K_{MF} \| V_3)$
$C^*_{MF} = h(h(K_{MF}) \| V_3)$
Check if $C'_{MF} = C_{MF}$
Store $C^*_{MF}$ and $V_3$

Figure 2: Authentication Phase

and computes the corresponding pseudo-identity $PID_{MU} = h(ID_{MU} \| b_2)$ for $MU$, and the value $SR = h(r \| s)$, where $s$ is the long-term secret of $HA$. It uses the hash of the user's password to compute $PSR = SR \oplus PW_{MU}$. Then, $HA$ computes and stores $DID_{MU} = PID_{MU} \oplus SR$, $VID_{MU} = h(r \| PID_{MU})$ in its database as shown in the Table 2. In the end, the home agent writes $r$, $b_2$, $PSR$ into a smart card and issues it to $MU$.

$$HA \rightarrow MU : SC = \{r, b_2, PSR\}.$$

Step 3. After receiving the smart card, the mobile user $ID_{MU}$ writes $b_1$ into it.

### 3.2 Authentication Phase

In this phase, $MU$ moves into a region handled by a foreign agent $FA$ whose identity is $ID_{FA}$. The mobile user $ID_{MU}$ will be authenticated anonymously by the home agent $HA$ as shown in the Figure 2 and the following steps:

Step 1. $MU$ inserts the smart card into the reader, and inputs its username $ID_{MU}$ and password $pw_{MU}$. The smart card uses the provided information to compute $SR = PSR \oplus PW_{MU} = h(r \| s)$, where $PW_{MU} = h(pw_{MU} \| b_1)$. Using the identity of the mobile user, the smart card computes the pseudo-$ID$ $PID_{MU} = h(ID_{MU} \| b_2)$, and the dynamic identity $DID_{MU} = PID_{MU} \oplus SR$. It then chooses a random number $r_1 \in \mathbb{Z}_p^*$, and computes $R_1 = r_1 \oplus SR$, and $V_1 = h(r_1 \| PID_{MU} \| ID_{FA})$, where $ID_{FA}$ is the identity of the current foreign agent. The smart card forms a message $m_1 = \{r, DID_{MU}, R_1, V_1, ID_{HA}\}$ and sends it to $FA$.

$$MU \rightarrow FA : m_1 = \{r, DID_{MU}, R_1, V_1, ID_{HA}\}.$$

Step 2. Upon receiving $m_1$, $FA$ chooses a random number $a \in \mathbb{Z}_p^*$ and computes $aP$. Then, it sends $m_2 = \{r, DID_{MU}, R_1, V_1, aP\}$ to $HA$.

$$FA \rightarrow HA : m_2 = \{r, DID_{MU}, R_1, V_1, aP\}.$$

Step 3. $HA$ uses its long-term secret $s$ to compute $SR' = h(r \| s)$, and then compute $PID'_{MU} = DID_{MU} \oplus SR'$, and $r' = R_1 \oplus SR'$. It looks up $DID_{MU}$

Table 2: $HA$'s database layout

| $Current - DID$ | $Current - VID$ | $Previous - DID$ | $Previous - VID$ |
|---|---|---|---|
| $DID_{MU}$ | $VID_{MU}$ | - | - |

in the database and retrieves the corresponding $VID_{MU}$. $HA$ checks whether $VID_{MU} = h(r\|PID'_{MU})$ and $V_1 = h(r'_1\|PID'_{MU}\|ID_{FA})$. If both equations hold, it trusts that $MU$ is valid. $HA$ then chooses a random number $r_2 \in \mathbb{Z}_p^*$ and computes $r* = r'_1 \oplus r_2$, $SR^* = h(r\|s)$, $R_2 = r_2 \oplus h(SR')$, $MSR = SR^* \oplus r_2$, $V_2 = h(r_2\|SR^*\|SR\|aP.x\|ID_{FA})$, and $V_3 = h(r\|r^*)$, where $aP.x$ is the $x$-axis value of the point $aP$.

$HA$ computes $PID^*_{MU} = h(r^*\|s)$, $DID^*_{MU} = PID^*_{MU} \oplus SR^*$, and $VID^*_{MU} = h(r^*\|PID^*_{MU})$. It saves the old values $DID_{MU}$ and $VID_{MU}$ to the $Previous - DID$ and $Previous - VID$ columns in the database, then writes $DID^*_{MU}$ and $VID^*_{MU}$ to the $Current - DID$ and $Current - VID$ columns, respectively. Finally, it transmits the message $m_3 = \{R_2, MSR, V_2, V_3\}$ to $FA$.

$$HA \rightarrow FA : m_3 = \{R_2, MSR, V_2, V_3\}.$$

Step 4. Ater receiving the confirmation from $HA$ that $MU$ is legitimate, $FA$ stores $V_3$ and forwards $R_2$, $MSR$, $V_2$, and $aP$ to $MU$.

$$FA \rightarrow MU : m_4 = \{R_2, MSR, V_2, aP\}.$$

Step 5. From $m_4$, $MU$'s smart card computes $r'_2 = R_2 \oplus h(SR)$, $SR^{*\prime} = MSR \oplus r'_2$. It verifies whether $V_2 \overset{?}{=} h(r'_2\|SR^{*\prime}\|SR\|aP.x\|ID_{FA})$. If it holds, the smart card believes that it is talking to a valid $FA$. It then computes $r^* = r_1 \oplus r'_2$, $V_3 = h(r\|r^*)$. It chooses a random number $b \in \mathbb{Z}_p^*$, and compute the session key $K_{MF} = h(abP.x)$, and $C_{MF} = h(K_{MF}\|V_3)$. It then sends $bP$, $C_{MF}$, to $FA$.

$$MU \rightarrow FA : m_5 = \{bP, C_{MF}\}.$$

After that, the smart card replaces the current $PSR$ in memory by $PSR^* = PW_{MU} \oplus SR^{*\prime}$, and $r$ by $r'$.

Step 6. Upon receiving $m_5$, $FA$ computes the session key $K_{MF} = h(abP.x)$, and verifies if $C_{MF} \overset{?}{=} h(K_{MF}\|V_3)$. If they are equal, $FA$ and $MU$ are mutually authenticated and share the session key $K_{MF}$. The foreign agent then computes $C^*_{MF} = h(h(K_{MF})\|V_3)$, and saves $C^*_{MF}$ and $V_3$ for this $MU$ in the database for roaming users.

At the end, the mobile user and the foreign agent are mutually authenticated and have a shared session key $K_{MF}$.

### 3.3 Session Key Update Phase

$MU$ and $FA$ can renew their session key so that $MU$ can extend its stay in the $FA$'s region or rejoin it. This phase commences with $MU$ sending $FA$ a session key update message as shown in Figure 3.

Step 1. $MU$'s smart card chooses a random number $c \in \mathbb{Z}_p^*$, and computes $C^*_{MF} = h(h(K_{MF})\|V_3)$, and $V_{MF1} = h(cP.x\|V_3)$, where $K_{MF}$ is the current session key. Then, it transmits $\{C^*_{MF}, cP, V_{MF1}\}$ to $FA$.

$$MU \rightarrow FA : m_6 = \{C^*_{MF}, cP, V_{MF1}\}.$$

Step 2. $FA$ searches for $C^*_{MF}$ in its database. If it is found, $FA$ trusts that $MU$ has been authenticated previously and it retrieves the corresponding $V_3$ from the database. After that, $FA$ verifies whether $V_{MF1} = h(cP.x\|V_3)$. If it holds, $FA$ chooses a random number $d \in \mathbb{Z}_p^*$ and computes the new session key $K^*_{MF} = h(cdP.x)$, and $V_{MF2} = h(V_3\|K^*_{MF})$. It then sends $\{dP, V_{MF2}\}$ to $MU$, and updates the current $C^*_{MF}$ with $C^*_{MF} = h(h(K^*_{MF})\|V_3)$.

$$FA \rightarrow MU : \{dP, V_{MF2}\}.$$

Step 3. $MU$ computes the new session key $K^*_{MF} = h(cdP.x)$ and checks if $V_{MF2} = h(V_3\|K^*_{MF})$. If the equation holds, $MU$ updates the session key to $K^*_{MF}$.

### 3.4 Password Changing Phase

In this phase, we suppose that $MU$ is already authenticated/ In order to update the password, $MU$ inputs the old password $pw_{MU}$ and the new password $pw^*_{MU}$. The smart card computes $PW_{MU} = h(b_1\|pw_{MU})$, and $PW^*_{MU} = h(b_1\|pw^*_{MU})$. At last, it replaces the old $PSR$ with $PSR^* = PSR \oplus PW_{MU} \oplus PW^*_{MU}$.

## 4 Security Analysis

Our scheme has the following security properties.

### 4.1 Mutual Authentication

In our scheme, we assume that $HA$ and $FA$ are mutually authenticated. In the authentication phase, $MU$ first authenticates with $HA$; then, $HA$ helps $FA$ and $MU$ to to authenticate each other. Our reasoning is based on BAN

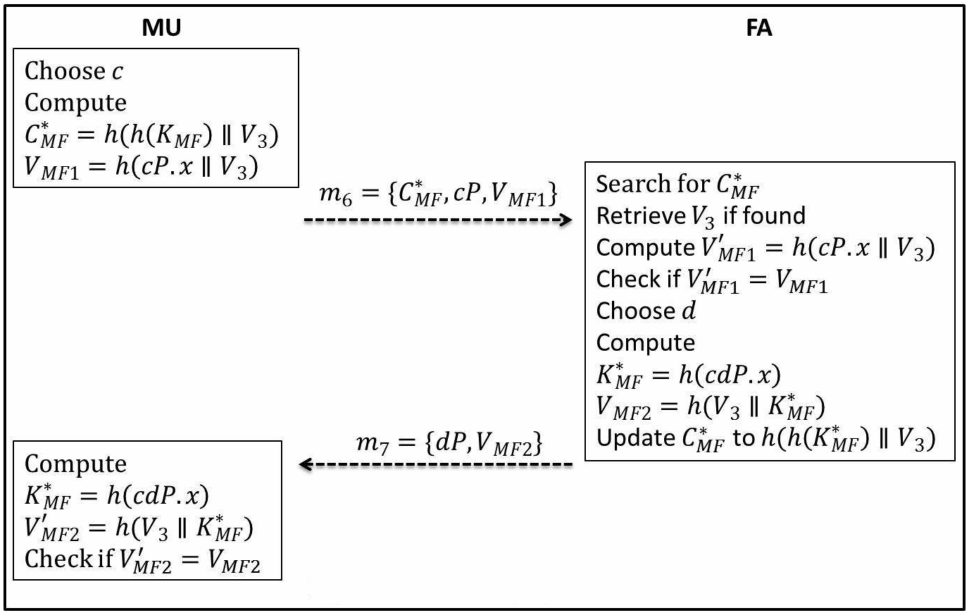

MU
FA
Choose $c$
Compute
$C^*_{MF} = h(h(K_{MF}) \| V_3)$
$V_{MF1} = h(cP.x \| V_3)$
$m_6 = \{C^*_{MF}, cP, V_{MF1}\}$
Search for $C^*_{MF}$
Retrieve $V_3$ if found
Compute $V'_{MF1} = h(cP.x \| V_3)$
Check if $V'_{MF1} = V_{MF1}$
Choose $d$
Compute
$K^*_{MF} = h(cdP.x)$
$V_{MF2} = h(V_3 \| K^*_{MF})$
Update $C^*_{MF}$ to $h(h(K^*_{MF}) \| V_3)$
$m_7 = \{dP, V_{MF2}\}$
Compute
$K^*_{MF} = h(cdP.x)$
$V'_{MF2} = h(V_3 \| K^*_{MF})$
Check if $V'_{MF2} = V_{MF2}$

Figure 3: Session key update phase

login [1] to prove that the proposed scheme provides mutual authentication for $MU - HA$ and $MU - FA$.

The value $r$ that is kept by $MU$ is *fresh* for each session, and $HA$ has a verifier of $r$ stored in its database under the tuple $Current - VID_{MU}$, where $VID_{MU} = h(r \| PID_{MU})$. When receiving $m_2$ from $FA$, $HA$ uses its long-term secret $s$ to derive $PID_{MU}$. Then, it verifies the received $r$ by competing $VID'_{MU}$ and comparing it with $VID_{MU}$. Since $HA$ trusts that $VID_{MU}$ is *fresh*, it believes that $m_2$ is also *fresh*. Therefore, it trusts that the parameters $r$, $DID_{MU}$, $R_1$, and $V_1$ come from the legitimate mobile user whose pseudo-$ID$ is $PID_{MU}$.

At the other end, $MU$ knows that its $r$ is *fresh*. Upon receiving $m_4$, it verifies $V_2 = h(r'_2 \| SR^{*\prime} \| SR \| aP.x \| ID_{FA})$. If $V_2$ is valid, $MU$ trusts that $V_2$ comes from $HA$ since only $HA$ can compute $SR = h(r \| s)$ using its long-term secret $s$. Because $ID_{FA}$ is in $V_2$, $MU$ also trusts that $FA$ is genuine. At this point, $MU$ and $HA$ believe that their counterpart is authentic. Since $HA$ helps $FA$ to authenticate $MU$, both $FA$ and $MU$ are also mutually authenticated.

## 4.2 Perfect Forward Secrecy

Our scheme uses ECDH (Elliptic Curve Diffie-Hellman) to provide perfect forward secrecy. In the computational ECDH problem, given two points $xP$ and $yP$, where $x$, $y \in \mathbb{Z}^*_p$, computing the point $xyP$ is infeasible. Therefore, ECDH is commonly used in establishing session key between two entities. Since $x$ and $y$ are selected at random for each session, there is no feasible way to compute the past session key $xyP$ without knowing either $x$ or $y$.

In the proposed scheme, $HA$ and $FA$ compute the session key $K_{MF} = h(abP.x)$, where $a$, $b$ are generated *freshly* for each session. It is infeasible to derive either $a$ or $b$ from $aP$ and $bP$, respectively. Based on ECDH, it is a computationally difficult problem to guess $abP$ provided $aP$ and $bP$. Therefore, no adversary can guess the session key $K_{MF}$ for any session even after compromising long-term secrets of $MU$ and $HA$.

## 4.3 Achieve Anonymity

In our protocol, $HA$ authenticates $MU$ by verifying its $PID_{MU} = h(ID_{MU} \| b_2)$. The parameters in the message $m_2$ from $FA$ to $HA$ do not contain $ID_{MU}$, but only carrying its hash value $PID_{MU}$ in $DID_{MU}$. Since a secure hash function is used, $HA$ cannot deduce $ID_{MU}$ from $PID_{MU}$.

Moreover, $ID_{MU}$ is never sent in plaintext. Only $PID_{MU}$ is sent over insecure channel in $DID_{MU} = PID_{MU} \oplus h(r \| s)$. $FA$ or an eavesdropping adversary $\mathcal{A}$ will not be able to derive $ID_{MU}$ from $DID_{MU}$. Therefore, the mobile user is anonymous to $HA$, $FA$, and $\mathcal{A}$.

## 4.4 Achieve Untraceability

At the end of each authentication phase, $r$ is assigned a new value $r^* = r_1 \oplus r_2$, where $r_1$, $r_2$ are *freshly* generated by $MU$ and $HA$, respectively; thus, it leads to the charges of $h(r \| s)$ and $DID_{MU} = PID_{MU} \oplus h(r \| s)$. Consequently, the parameters in the messages originated from $MU$ do not retain the same values for different sessions. Therefore, $FA$ and $\mathcal{A}$ cannot know whether the messages in two different sessions come from the same $MU$ or not.

Table 3: Comparison regarding security properties

| | Mun et al. [7] | Xie et al. [10] | Kuo et al. [5] | Ours |
|---|---|---|---|---|
| Achieve anonymity | Yes | Yes | Yes | Yes |
| Achieve untraceability | Yes | Yes | Yes | Yes |
| Provide perfect forward secrecy | Yes | Yes | Yes | Yes |
| Prevent disclosure of user's password | No | No | Yes | Yes |
| Prevent replay attack | No | Yes | Yes | Yes |
| Provide mutual authentication $(MU-HA)$ | Yes | Yes | Yes | Yes |
| Provide mutual authentication $(MU-FA)$ | Yes | Yes | Yes | Yes |
| Prevent man-in-the-middle attack | Yes | No | No | Yes |
| Session key security | Yes | Yes | Yes | Yes |
| Smart card lost attack | No | Yes | No | No |
| Stolen verifier attack | No | No | Yes | No |

### 4.5 Prevent Disclosure of User's Password

In the proposed scheme, the mobile user's password is only used to compute $PW_{MU} = h(pw_{MU}\|b_1)$, and there is no information related to $PW_{MU}$ in any message sent from $MU$. Therefore, no adversary can obtain $pw_{MU}$ by eavesdropping on the communication channel.

### 4.6 Prevent Man-in-the-middle Attack

Kuo et al.'s suffers four kinds of man-in-the-middle attacks because it does not verify $aP$ and $bP$ properly. In our scheme, $MU$ makes sure that $aP$ comes from $FA$, and $FA$ can verify that $bP$ comes from $MU$.

$HA$ provides $FA$ with $V_3 = h(r\|r^*)$ so that it can authenticate $bP$. When receiving $m_4$ from $MU$, $FA$ computes the session key $K_{MF}$ and uses $V_3$ to verify it by comparing $h(K_{MF}\|V_3)$ against the received value $C_{MF}$.

At the mobile user side, the smart card can verify $aP$ since $aP.x$ is contained in $V_2 = h(r_2\|SR^*\|SR\|aP.x\|ID_{FA})$ which is sent from $HA$. If $MU$ and $HA$ are already mutually authenticated, $MU$ will trust that $aP$ indeed comes from $FA$.

### 4.7 Prevent Replay Attack

The values $r$, $r_1$, $a$, and $b$ are generated for each session, and the parameters in all the messages are all related to them, Those values are verified by $MU$, $FA$, and $HA$; therefore, an adversary cannot deceive any legitimate entity by replaying old messages.

### 4.8 Prevent Stolen Verifier Attack

If an attacker $\mathcal{A}$ compromises $HA$'s database, it can obtain $DID_{MU} = PID_{MU} \oplus h(r\|s)$, and $VID_{MU} = h(r\|PID_{MU})$. However, without $HA$'s long-term secret $s$, it cannot compete $h(r\|s)$ which is used in computing $R_2$, $V_2$. Therefore, it will not be able to masquerade as $FA$ like it could do in Kuo et al.'s scheme.

### 4.9 Prevent Smart Card Lost Attack

If $\mathcal{A}$ obtains a valid smart card of $ID_{MU}$, it can retrieve $b_1$, $b_2$, $r$ and $PSR = h(r\|s) \oplus PW_{MU}$, where $PW_MU = h(pw_{MU}\|b_1)$. The attacker does not know $pw_{MU}$ to compute $PW_{MU}$. Without the password, $\mathcal{A}$ cannot derive $SR = h(r\|s)$ which plays essential role in authentication. Therefore, loosing smart card will not compromise the security of the system.

## 5 Functionality and Performance Analysis

In this section, we evaluate our proposed scheme in terms of security properties and computation costs. We compare these features in our scheme with their counterparts in Mun et al.'s [7], Xie et al's [10], and Kuo et al's [5].

The comparison for security features is shown in the Table 3. All the schemes can achieve mutual authentication ($MU-HA$, $MU-FA$), perfect forward secrecy, session key security, anonymity, and untraceability. However, Mun et al.'s and Xie et al.'s schemes are vulnerable to disclose the mobile user's password. Both Xie et al.'s and Kuo et al.'s cannot prevent man-in-the-middle attacks. Since there is no verifier database in Mun et al.'s and Xie et al.'s, stolen verifier attack is not a threat to them, but Kuo et al.' scheme is susceptible to this kind of attack. And lastly, all the schemes except Xie et al.'s are immune to smart card lost attack.

The schemes' performances in term of computation costs in the authentication phase are shown in Table 4. Mum et al.'s and Xie et al.'s employ both symmetric and asymmetric cryptography in their schemes. Therefore, their computing workloads are higher than Kuo et al.'s and our scheme. Like in Kuo et al.'s, we use only elliptic curve point multiplications in establishing session key. $MU$ in our scheme performs less computing than Kuo et al.'s one, whereas computing workloads on our $FA$ and $HA$ are slightly higher than Kuo et al.'s. However, our scheme is more sufficient than Kuo et al.'s in the pass-

Table 4: Comparison regarding computation costs

| | Mun et al. [7] | Xie et al. [10] | Kuo et al. [5] | Ours |
|---|---|---|---|---|
| $MU$ | $2t_p + t_s + 5t_h + 2t_{XOR}$ | $3t_e + 4t_h + 2t_s + t_{XOR}$ | $2t_p + 9t_h + 6t_{XOR}$ | $2t_p + 7t_h + 7t_{XOR}$ |
| $FA$ | $2t_p + t_s + 4t_h + 2t_{XOR}$ | $3t_e + 2t_h + 3t_s$ | $2t_p + 2t_h$ | $2t_p + 3t_h$ |
| $HA$ | $5t_h + 3t_{XOR}$ | $2t_e + t_h + 4t_s + t_{XOR}$ | $6t_h + 2t_{XOR}$ | $8t_h + 6t_{XOR}$ |
| $Total$ | $4t_p + 2t_s + 14t_h + 7t_{XOR}$ | $8t_e + 7t_h + 9t_s + 2t_{XOR}$ | $4t_p + 11t_h + 8t_{XOR}$ | $4t_p + 18t_h + 13t_{XOR}$ |

$t_e$ : time for performing modular exponentiation
$t_p$ : time for performing elliptic curve point multiplication
$t_s$ : time for performing symmetric encryption/decryption
$t_h$ : time for performing hash operation
$t_{XOR}$ : time for performing XOR operation

word changing phase as shown in Table 5 since $HA$ does not have to involve in the process.

Table 5: Computing workloads in Password Changing Phase

| | Kuo et al. [5] | Ours |
|---|---|---|
| $MU$ | $4t_h + 2t_{XOR}$ | $2t_h + 2t_{XOR}$ |
| $FA$ | 0 | 0 |
| $HA$ | $2t_h + t_{XOR}$ | 0 |

# 6 Conclusions

In this paper, we proposed a novel and secure authentication and key agreement scheme for roaming service in global mobile network. Our scheme achieves mutual authentication for $MU - HA$, and $MU - FA$. To ensure mobile user's anonymity, pseudo-identity is used in place of the actual identity. All the parameters in the messages exchanged are *fresh* and not repeated so that mobile user's activities are not traceable. Perfect forward secrecy is preserved even in the extreme case where the long-term secret of $HA$ is compromised. Furthermore, the proposed scheme uses mostly hash functions and XOR operations, and very few elliptic curve point multiplication; as a result, it is very efficient and suitable for use in mobile networks.

# Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

# References

[1] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 426, pp. 233–271. The Royal Society, 1989.

[2] C. C. Chang, C. Y. Lee, and Y. C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.

[3] D. He, N. Kumar, M. Khan, and J. H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 811–817, 2013.

[4] J. S. Kim and J. Kwak, "Improved secure anonymous authentication scheme for roaming service in global mobility networks," *International Journal of Security and Its Applications*, vol. 6, no. 3, pp. 45–54, 2012.

[5] W. C. Kuo, H. J. Wei, and J. C. Cheng, "An efficient and secure anonymous mobility network authentication scheme," *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 18–24, 2014.

[6] C. C. Lee, M. S. Hwang, and I-E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683–1687, 2006.

[7] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1, pp. 214–222, 2012.

[8] S. Suzuki and K. Nakada, "An authentication technique based on distributed security management for the global mobility network," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 8, pp. 1608–1617, 1997.

[9] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722–723, 2008.

[10] Qi Xie, B. Hu, X. Tan, M. Bao, and X. Yu, "Robust anonymous two-factor authentication scheme for roaming service in global mobility network,"

*Wireless personal communications*, vol. 74, no. 2, pp. 601–614, 2014.

[11] C. K. Yeh and W. B. Lee, "An overall cost-effective authentication technique for the global mobility network," *International Journal of Network Security*, vol. 9, no. 3, pp. 227–232, 2009.

[12] T. Y. Youn, Y. H. Park, and J. Lim, "Weaknesses in an anonymous authentication scheme for roaming service in global mobility networks," *IEEE Communications Letters*, vol. 13, no. 7, pp. 471–473, 2009.

[13] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 231–235, 2004.

**Hai-Duong Le** received his B.E. degree in 2004 at University of Tasmania, Australia, and his M.I.T degree in 2006 at James Cook University, Australia. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from Feng Chia University, Taichung, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

**Chin-Chen Chang** received his Ph.D. degree in computer engineer- ing from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visit- ing Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organisations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.

**Yeh-Chieh Chou** was born in Taichung, Taiwan, in 1990. He received his Bachelor?s Degree in Information Engineering form Chang Jung Christian University, Tainan. Currently, he is the second grade student for master?s program of Department of Information Engineering in Feng Chia University. His research interests include computer cryptography and information security.

# EA Based Dynamic Key Generation in RC4 Ciphering Applied to CMS

Ashraf Aboshosha[1], Kamal A. ElDahshan[2], Eman K. Elsayed[3], and Ahmed A. Elngar[2]

*(Corresponding author: Ahmed A. Elngar)*

NCRRT, Atomic Energy Authority, Cairo, Egypt[1]
Faculty of Science, Al-Azhar University, Cairo, Egypt[2]
(Email: elngar_7@yahoo.co.uk)
Faculty of Science (Girls), Al-Azhar University, Cairo, Egypt[3]

*(Received Feb. 8, 2015; revised and accepted Mar. 25 & Apr. 21, 2015)*

## Abstract

RC4 is the most widely used stream cipher algorithm. It is used to protect valuable electronic information. However, RC4 encryption algorithm suffers from a secret key generation as a seed problems. This paper proposes an intelligent dynamic secret key generation as a seed by employing an Evolutionary Algorithm (EA). The proposed RC4-EA method tends to enhance the RC4 encryption algorithm with a high degree of a seed key randomness. The main advantage of the proposed RC4-EA method is that the generation of this secret key is done dynamically and randomly; this adds more strength of the RC4 encryption algorithm against breaking this cryptosystems. Several experiments on the proposed RC4-EA method are conducted. Where, the results of the experiments show the improvement of the encryption time and the throughput of the proposed encryption RC4-EA method. Also, the proposed validated RC4-EA encryption method is applied for data ciphering in Content Management System (CMS).

*Keywords: Confidentiality, evolutionary algorithm (EA), RC4*

## 1 Introduction

During the past decades, Internet Technology (IT) has influenced everyday life [5, 23]. Internet security issues have become more common nowadays; particularly on internet banking account, shopping online and content management systems (CMSs) due to the harmful impact on confidentiality, integrity and privacy [21]. Sensitive information should stay secured and such security should be transparent and ubiquitous whenever shared [24]. Cryptography plays a major role in helping to prevent eavesdropping of sensitive information [7]. Encryption is the process of transforming plaintext into cipher text in order to prevent any unauthorized recipient from retrieving the original data [10]. The encrypted data is sent over the public network and is decrypted by the intended recipient [25].

One of the cryptographic algorithms is RC4 stream cipher algorithm. RC4 is considered to be a perfect cipher algorithm, but it suffers from some drawbacks [1]. One of the main drawbacks is that; a nonrandom secret key $k$ as a seed is exposed to the attacker [6, 22]. Since the same secret key when permutes with the exposed Initial Vector (IV), an attacker can re-derive the secret key by analyzing the initial words of the key streams with relatively little work [18].

The random number generator is deterministic and periodic, which means that the sequence of numbers will eventually repeat itself or reproduced at later date. So the random number generator is not suitable for applications where it is important that the numbers are really unpredictable, such as data encryption.

Evolutionary Algorithm (EA), which is based on a powerful principle of evolution: survival of the fittest which model the natural phenomena [3]. EA has been widely used in science for solving complex problems [4].

In this paper, a stream cipher RC4 employees an Evolutionary Algorithm (EA) based approach to generate a dynamic secret key as a seed permutes with Initial Vector (IV) to produce a final key stream for encrypting the data is proposed. The novelty in the proposed method is that; EA is used to generate a dynamic random secret keys as a seed used for RC4 encryption algorithm which leads to increase the security of the system.

In the proposed RC4-EA method, the plaintext is encrypted in the form of ciphertext. Where, EA based approach is used to generate the dynamic random secret key based on a normal biological evolution. Since we get the key it is sent securely to RC4 encryption algorithm which permute with the (IV) to generate the the final key stream. An XOR operation is performed on the final key stream with the plaintext to obtain the ciphertext then storzed in the database.

The advantage of the proposed RC4-EA method is to increase the security of the system, by generating the secret keys dynamically and randomly. Which leads to, overcome the drawback of a non-random secret key as a seed in the original RC4 encryption algorithm. Hence, the final key stream can not be cracked by the attacker.

The rest of this paper is organized as follows: Section 2 presents an overview of the used algorithms, including the RC4 Encryption algorithm, weaknesses and attacks over RC4, and Evolutionary Algorithm (EA). Section 3 introduces the proposed dynamic RC4-EA encryption method. Section 4 gives the implementation results and analysis. Section 5 presents an applicable case study of the proposed RC4-EA encryption method. Finally, Section 6 contains the conclusion remarks.

# 2 An Overview

## 2.1 RC4 Encryption Algorithm

RC4 is a stream cipher algorithm designed in 1987 by Ron Rivest for RSA Security. The details remained secret until 1994, when they were anonymously published on an internet newsgroup.

RC4 is the most widely used software stream cipher in the world. It is used to protect internet traffic as part of the SSL (Secure Socket Layer), TLS (Transport Layer Security) protocols [20]. RC4 also used for encryption in the Wired Equivalent Privacy (WEP) (part of the IEEE 802.11), and Wi-Fi Protected Access (WPA) (part of the IEEE 802.11 i) protocols [14].

The RC4 algorithm consists of a permutation of array $S$ containing the numbers $0, ...., N-1$, and two indices $i$, $j$ in the initial state, where $N = 256$ . The algorithm can be broken into two main stages: the Key Scheduling Algorithm (KSA), which uses the secret key $k$ as a seed to create a pseudo-random initial state, and the Pseudo Random Generation Algorithm (PRGA), which generates the pseudo-random stream [19]. The flow chart of the RC4 encryption algorithm is shown in Figure 1.

### 2.1.1 Initialization

At the internal state, the entries of array $S$ are set equal to the values $0, ..., N-1$ in ascending order; that is $S[0] = 0, S\ [1] = 1, ...\ , S[255] = 255$. Also, a temporary Vector $T$ is created. If the length of the secret key $k$ is 256 bits, then $k$ is transferred to $T$. Otherwise, for a secret key $k$ of length $\ell$ bits, the first element is copied from $k$ to $T$ till the length $\ell$, then $k$ is repeated as many times as necessary to fill out $T$ [8].

```
$for\ \ i\ = \ 0\ \ to\ \ 255\ \ do$
    $S[i] = i$ ;
      $T[i] = k\ [\ i\ \ mod\ \ \ell\ ]$;
```

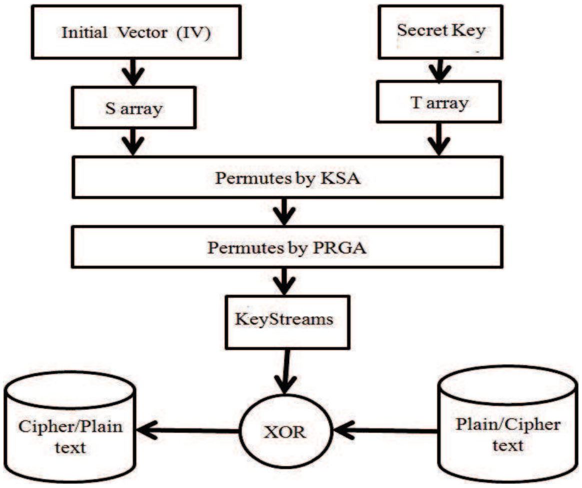


Figure 1: Encryption and decryption by RC4

### 2.1.2 The Key Scheduling Algorithm

The Key Scheduling Algorithm (KSA) is used to generates a pseudo-random initial permutation of array $S$. Once the $S$ array is initialized, $S$ is shuffled using the secret key $T[i]$ to make it a permutation array [9]. The following actions are iterated 256 times after initializing $i$ and $j$ to 0:

```
$Compute\ \ j = \ (\ j\ +\ S[i]\ +\ T[i]\ )\ \ mod\ \ 256;$
$Swap\ \ (\ S[i],\ S[j]\ );$
  $increment\ \ \ i;$
```

Once $i$ has reached 256, the $S$ array has been properly initialized [19]. The Key Scheduling Algorithm (KSA) is given in Algorithm 1.

**Algorithm 1** A key scheduling algorithm (KSA)

```
Input:
$S\ \ \ \ //before\ \ permutation$
$T\ //A\ temporary\ vector\ of\ secret\ key\ k\ as\ a\ seed$
Output:
$array\ \ S\ \ \ \ \ //after\ \ permutation$

for i = 0 to 255 do
    S[i] = i
end for
$j = 0$
for i = 0 to 255 do
    $j =\ (\ j + S[i] + T[i]\ )\ \ mod\ \ 256$
    $Swap\ \ (\ S[i],\ S[j]\ )$
end for
Return (S)
```

### 2.1.3 The Pseudo-Random Generation Algorithm (PRGA)

Once the $S$ vector is initialized, the input secret key is no longer used [7]. The PRGA algorithm can generate key streams of any size. First, it initializes the two indexes

$i$, $j$ to 0 and then starts the stream generation with $S[0]$ till $S[255]$. For each $S[i]$, $S[i]$ are swapped with $S[j]$ according to the following actions [12]:

1) Compute new value of i and j.
$i = ( i+1 )\ mod\ 256;$
$j = ( j+S[i] )\ mod\ 256;$

2) Swap $S[i]$ and $S[j]$ to have a dynamic state (it makes it harder to crack than if the state was computed only once and use for the generation of the whole key streams).
$Swap\ ( S[i],\ S[j] );$

3) Retrieve the next byte of the key stream from the array $S$ at the index $u$.
$u = S[(S[i] + S[j])\ mod\ 256];$

The algorithm of the PRGA is given in Algorithm 2.

**Algorithm 2** Pseudo-random generation algorithm (PRGA)

```
Input:
$S$    // $State\ of\ array\ S$
$u$              //$A\ temporary\ vector$
Output:
$K$               //$sequence\ of\ keystreams$

$i = 0$
$j = 0$
while $not\ end\ of\ sequence$ do
    $i = ( i+1 )\ mod\ 256$
    $j = ( j+S[i] )\ mod\ 256$
    $Swap\ ( S[i],\ S[j] )$
    $u = S[(S[i] + S[j])\ mod\ 256]$
      $K = S[u]$
end while
Return (K)
```

### 2.1.4 Encryption and Decryption

Once the final key stream has been generated, the encryption and decryption process is the same as, the plaintext is XORed with the generated final key stream. If it is fed in plaintext, it will produce the cipher text, and if it is fed in a cipher text, it will produce the plaintext output [15].

## 2.2 Weakness And Attacks Over RC4

The cryptanalysis of the RC4 algorithm was divided into two parts, (1) analysis of the initialization of RC4 which focuses on the initialization of KSA, and (2) analysis of the output key streams generation which focuses on the internal state and the round operation of PRGA. To make the RC4 algorithm secure and capable to stand against attacks, lot of research are done over RC4 algorithm to enhancing its security.

**Pardeep and Pushpendra** in [16], reported that the RC4 algorithm was disclosed to the market and then experts start to analyze the RC4 algorithm and find out lots of weaknesses in both of two main stages of the algorithm KSA and PRGA.

**Mantin and Shamir** in [13], find out the weakness in the second round, where the probability of Zero output bytes are the major weakness of the RC4 algorithm.

**Fluher et al.** in [6], discovered the big weakness in the RC4, if anyone know the portion of the secret key then possible attacks fully over RC4.

**Paul and Maitra** in [17], generate the secret key by using the initial state table. They generated some equation on the bases of initial state table and they select some of the bytes of secret key on the bases of guess and the remain secret key find out by using the equation.

As the security of RC4 algorithm depends on the security of the secret key and the internal states of array $S$, thus many attacks focus on resuming the secret key of the internal states of the array $S$.

## 2.3 Evolutionary Algorithm

In real world applications Evolutionary Algorithm (EA), offers practical advantages to the researchers from facing difficult optimization problems [26]. These advantages are manifold, including the simplicity of the algorithm, its robust response to change circumstance, and its flexibility [2].

The EA can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. One of these problems is to implement diverse high-quality Random Number Generators (RNGs). As a result, EA has recently received increased interest, particularly with regard to the manner in which it may be applied for practical problem solving [11].

The conceptual base of EA to simulate the evolution of individual structures is via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by the problem [2].

**Algorithm 3** Evolutionary algorithm (EA)

```
Starting with the source string.
Initialize a population of random parents.
Evaluate the population of the parents based on a
  fitness function.
while The parent is not yet the target. do
    Apply mutation to the selected parents.
    Evaluate the fitness of all parents in the population,
    and keep the most fit string as the new parent
    repeat until the fit parent converges to the target.
end while
Return the number of iteration of the best parent.
```

Algorithm 3 is starting with the Source string . A population of a random parents are initialized, then new parent is created by applying reproduction operator (mutation). The fitness of the resulting solution is evaluated towards the Target string. Then, a suitable selection strategy is applied to determine which parent is maintained into the next generation. The process is iterated until a candidate parent with sufficient quality is found. Finally, it returns the number of iteration of the best string closed to the Target string.

# 3 Proposed Dynamic RC4-EA Encryption Method

A dynamic RC4-EA method is used for encrypting and decrypting the plaintext. Where, the EA algorithm is adapted to generate a dynamic secret key as a seed used in the RC4 encryption algorithm. Then, XOR operation is performed with the final key stream generated from the RC4-EA method on the plaintext to obtain the ciphertext, which is then stored in the database. The proposed RC4-EA method is divided into the following phases.

## 3.1 Generate Dynamic Secret Key Phase

To generate the dynamic secret key, EA will start with some characters which can be view as string. It will randomly mutate these characters to evolve the string toward the target. The following fitness function is used to judge the new mutated string fitness.

```
 Fitness (source, target)
  fitval = 0 .
   for i = 0 to len (source)
     fitval += (ord(target[i]) − ord(source[i]))$^2$ .
       return (fitval) .
```

The mutation function is given below, where only one character is mutated by one value at a time.

```
 mutate (source)
charpos = rand ( 0 , len ( source ) − 1) .
 parts = list ( source ) .
  parts [charpos] = char ( ord (part [ charpos ])
    + rand ( 0 , len (source − 1))) .
      return (' ' .join ( parts ) ) .
```

Once reaching the best string; the number of iteration of the best string is used as the random secret key after multiply it $n$ times to obtain the desirable length. Then this secret key is passed to the encryption plaintext phase.

The chart of an EA is illustrated in Figure 2

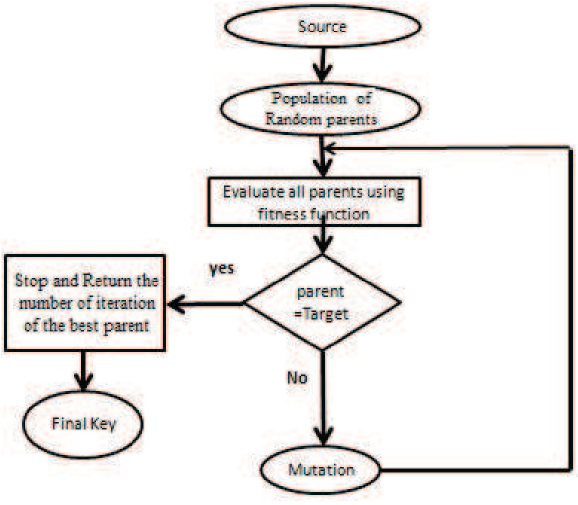

Source
Population of Random parents
Evaluate all parents using fitness function
yes
Stop and Return the number of iteration of the best parent
parent =Target
No
Final Key
Mutation

Figure 2: Generate dynamic secret key phase

## 3.2 Encryption Plaintext Phase

Figure 3 shows the proposed RC4-EA method for encrypting the plaintext. Where, the dynamic secret key is permute with the initial value (IV) in RC4 algorithm to generate the final key stream for encrypting the plaintext. To ensure the security, the sequence of key streams obtained is never used more than once. Instead of using the fixed secret key to generate key streams; we generate it dynamically and randomly. The randomness of the secret key will enhance the security of the system and prevent a hacker to crack final key stream which XORed with the plaintext. As we get the ciphertext, it is stored in the database.

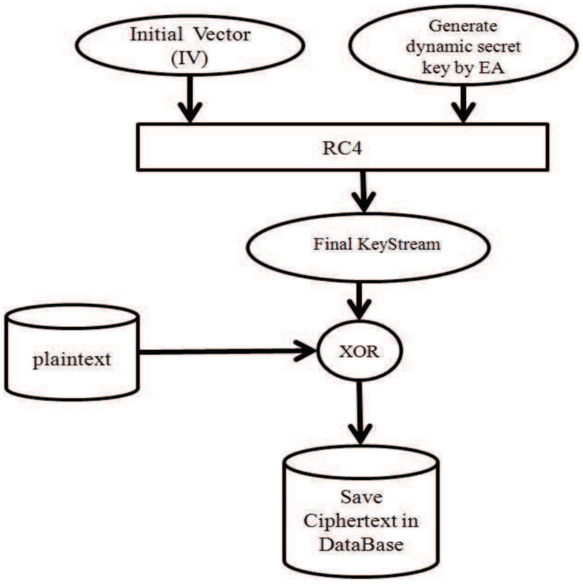

Initial Vector (IV)
Generate dynamic secret key by EA
RC4
Final KeyStream
plaintext
XOR
Save Ciphertext in DataBase

Figure 3: Encryption plaintext phase

# 4 Implementation Results and Analysis

The proposed RC4-EA encryption method is used for encrypting and decrypting the plaintext. All experiments have been performed using server 32 core AMD Opteron processor 6376 with 32 GB of RAM and 4 RAID 1s. The experiments have been implemented using PHP-MySql language environment.

### 4.1 Performance Evaluation

For the experiments, the performance evaluation of the encryption and decryption process is measured on different file sizes ranging from 20KB to 100KB. The performance evaluation metrics are:

1) **Encryption time:** It is the time that an encryption method takes to produce a ciphertext from a plaintext. As the encryption time decreases; the performance of the method increases.

2) **Throughputs:** The throughput of the encryption method is calculated as the total plaintext encrypted in $KB$ divided by the encryption time in microseconds. As the throughput increases, the performance increases and the power consumption decreases.

### 4.2 Experiments and Analysis

To analyze the performance of the proposed RC4-EA encryption method, a performance comparison between the original RC4 encryption algorithm and the proposed RC4-EA method is conducted. Based on the key length, two different cases of experiments are evaluated:

1) With key length 128 bits and data size ranging from 20KB to 100KB.

2) With key length 256 bits and data size ranging from 20KB to 100KB.

**Case 1: Key length 128 bits**
The average encryption times in $\mu s$ of the proposed RC4-EA encryption method and the original RC4 encryption algorithm were calculated over 10 different (random) key of length 128 bits. Where, the average encryption times are measured on different plaintext data sizes ranging from 20KB to 100KB. The comparison evaluation are shown in Table 1.

Table 1: Encryption time vs. data size with key length 128 bits

| Data Size (KB) | Encryption Time RC4 ($\mu s$) | Encryption Time RC4-EA ($\mu s$) |
|---|---|---|
| 20 | 1037.1208 | 905.9906 |
| 40 | 1085.9966 | 940.0845 |
| 60 | 1156.0917 | 978.9467 |
| 80 | 1192.0929 | 982.0461 |
| 100 | 1219.0342 | 988.9603 |

Table 2 shows the throughputs of the proposed RC4-EA encryption method compared with the original RC4 encryption algorithm.

From the results obtained with key length 128 bits, it is clear that the proposed RC4-EA encryption method shows an enhance in the encryption times and the throughputs compared to the original RC4 encryption algorithm, as shown in Figure 4 and Figure 5.

Table 2: Throughputs vs. data size with key length 128 bits

| Data Size (KB) | Throughput RC4 ($KB/S$) | Throughput RC4-EA ($KB/S$) |
|---|---|---|
| 20 | 19284.15 | 22075.28 |
| 40 | 36832.52 | 42549.36 |
| 60 | 51898.99 | 61290.36 |
| 80 | 67108.86 | 81462.57 |
| 100 | 82032.15 | 101116.29 |

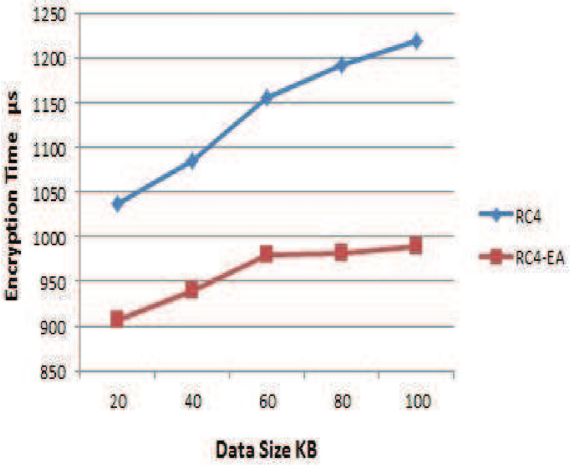


Figure 4: Encryption time of different data size with secret key of length 128 Bits

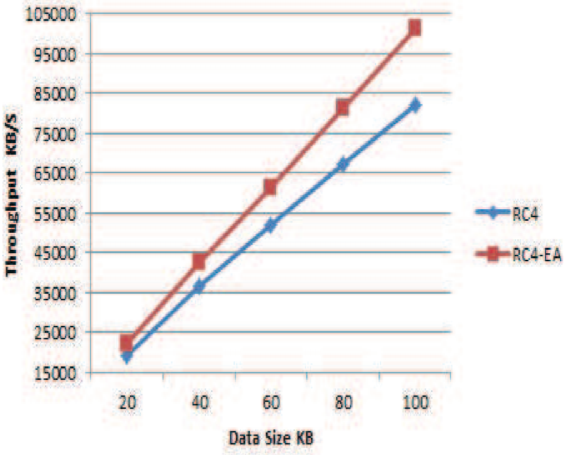


Figure 5: Throughputs for the encryption scheme of different data size with secret key of length 128 bits

**Case 2: Key length 256 bits**
Table 3 shows the average encryption times in $\mu s$ of the proposed RC4-EA encryption method and the original RC4 encryption algorithm, with 10 different (random) key

of length 256 bits. The encryption time is calculated for different plaintext data size varying from 20KB to 100KB.

Table 3: Encryption time vs. data size with key length 256 bits

| Data Size KB | Encryption Time RC4 ($\mu s$) | Encryption Time RC4-EA ($\mu s$) |
|---|---|---|
| 20 | 1105.0701 | 1036.9219 |
| 40 | 1125.0973 | 1044.9886 |
| 60 | 1189.9471 | 1047.1344 |
| 80 | 1214.9811 | 1052.8564 |
| 100 | 1260.0422 | 1065.9695 |

The throughputs of the proposed RC4-EA encryption method compared with the original RC4 encryption algorithm with key length 256 bits are given in Table 4.

Table 4: Throughputs vs. data size with key length 256 bits

| Data Size (KB) | Throughput RC4 ($KB/S$) | Throughput RC4-EA ($KB/S$) |
|---|---|---|
| 20 | 18098.39 | 19287.85 |
| 40 | 35552.48 | 38277.92 |
| 60 | 50422.4 | 57299.23 |
| 80 | 65844.64 | 75983.77 |
| 100 | 79362.42 | 93811.31 |

From Figures 6 and 7, it is clear that the experiments results with key length 256 bits, show an enhance to the proposed RC4-EA encryption method for the encryption times and the throughputs compared to the original RC4 encryption algorithm.

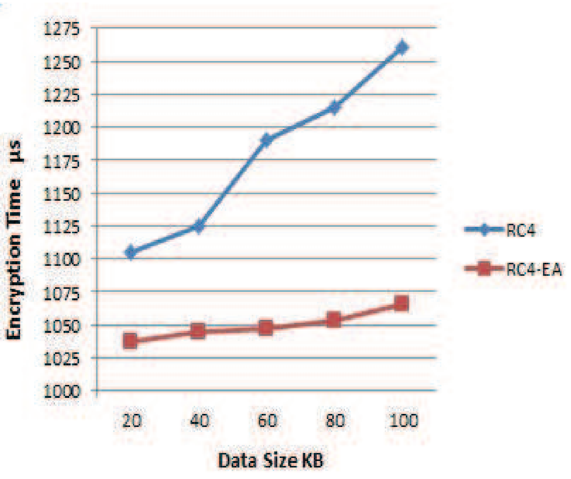


Figure 6: Encryption time of different data size with secret key of length 256 bits

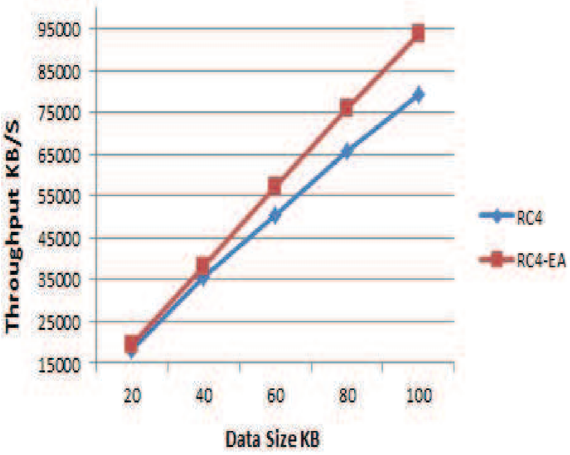
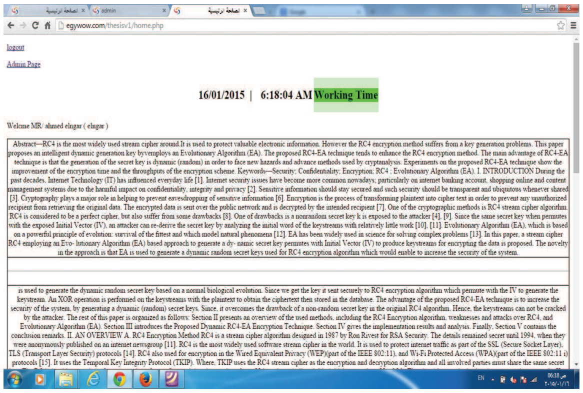


Figure 7: Throughputs for the encryption method of different data size with secret key of length 256 bits

## 5 Applicable Case Study Using RC4-EA Encryption Technique

The proposed RC4-EA encryption method is applied for data ciphering in Content Management System (CMS) in order to keep the data in high confidential authentication. Where the proposed RC4-EA method is used during the development of the web site (www.egywow.com/thesisv1).

Figure 8 and Figure 9 are samples of data in form of plaintext,and the same data in ciphertext form in (CMS) respectively .



16/01/2015 | 6:18:04 AM Working Time

Welcme MR/ ahmed elngar ( elngar )

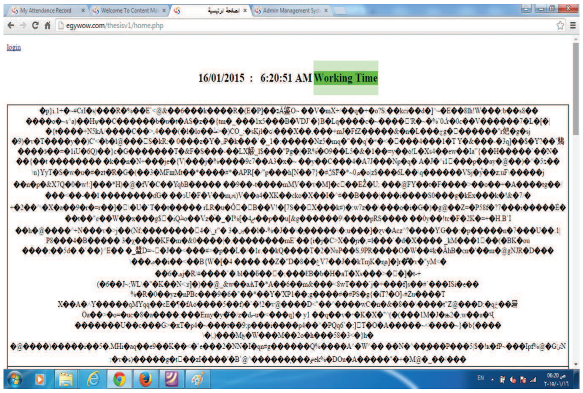
Abstract—RC4 is the most widely used stream cipher around.It is used to protect valuable electronic information. However the RC4 encryption method suffers from a key generation problems. This paper proposes an intelligent dynamic generation key byvemploys an Evolutionary Algorithm (EA). The proposed RC4-EA technique tends to enhance the RC4 encryption method. The main advantage of RC4-EA technique is that the generation of the secret key is dynamic (random) in order to face new hazards and advance methods used by cryptanalysis. Experiments on the proposed RC4-EA technique show the improvement of the encryption time and the throughputs of the encryption scheme. Keywords—Security; Confidentiality; Encryption; RC4 ; Evolutionary Algorithm (EA). I. INTRODUCTION During the past decades, Internet Technology (IT) has influenced everyday life [1]. Internet security issues have become more common nowadays; particularly on internet banking account, shopping online and content management systems due to the harmful impact on confidentiality, integrity and privacy [2]. Sensitive information should stay secured and such security should be transparent and ubiquitous whenever shared [3]. Cryptography plays a major role in helping to prevent eavesdropping of sensitive information [6]. Encryption is the process of transforming plaintext into cipher text in order to prevent any unauthorized recipient from retrieving the original data. The encrypted data is sent over the public network and is decrypted by the intended recipient [7]. One of the cryptographic methods is RC4 stream cipher algorithm. RC4 is considered to be a perfect cipher, but also suffer from some drawbacks [8]. One of drawbacks is a nonrandom secret key k is exposed to the attacker [4], [9]. Since the same secret key when permutes with the exposed Initial Vector (IV), an attacker can re-derive the secret key by analyzing the initial word of the keystreams with relatively little work [10], [11]. Evolutionary Algorithm (EA), which is based on a powerful principle of evolution: survival of the fittest and which model natural phenomena [12]. EA has been widely used in science for solving complex problems [13]. In this paper, a stream cipher RC4 employing an Evo- lutionary Algorithm (EA) based approach to generate a dy- namic secret key permutes with Initial Vector (IV) to produce keystreams for encrypting the data is proposed. The novelty in the approach is that EA is used to generate a dynamic random secret keys used for RC4 encryption algorithm which would enable to increase the security of the system.

is used to generate the dynamic random secret key based on a normal biological evolution. Since we get the key it sent securely to RC4 encryption algorithm which permute with the IV to generate the keystream. An XOR operation is performed on the keystreams with the plaintext to obtain the ciphertext then stored in the database. The advantage of the proposed RC4-EA technique is to increase the security of the system, by generating a dynamic (random) secret keys. Since, it overcomes the drawback of a non-random secret key in the original RC4 algorithm. Hence, the keystreams can not be cracked by the attacker. The rest of this paper is organized as follows: Section II presents an overview of the used methods, including the RC4 Encryption algorithm, weaknesses and attacks over RC4, and Evolutionary Algorithm (EA). Section III introduces the Proposed Dynamic RC4-EA Encryption Technique. Section IV gives the implementation results and analysis. Finally, Section V contains the conclusion remarks. II. AN OVERVIEW A. RC4 Encryption Method RC4 is a stream cipher algorithm designed in 1987 by Ron Rivest for RSA Security. The details remained secret until 1994, when they were anonymously published on an internet newsgroup [11]. RC4 is the most widely used software stream cipher in the world. It is used to protect internet traffic as part of the SSL (Secure Socket Layer), TLS (Transport Layer Security) protocols [14]. RC4 also used for encryption in the Wired Equivalent Privacy (WEP)(part of the IEEE 802:11), and Wi-Fi Protected Access (WPA)(part of the IEEE 802:11 i) protocols [15]. It uses the Temporal Key Integrity Protocol (TKIP). Where, TKIP uses the RC4 stream cipher as the encryption and decryption algorithm and all involved parties must share the same secret

Figure 8: Data before the encryption using the proposed RC4-EA encryption method

## 6 Conclusions

The major contribution of this paper is proposing RC4-EA encryption method for encrypting the plaintext. The proposed RC4-EA encryption method solves the deficiency of RC4 encryption algorithm which are caused by



16/01/2015 : 6:20:51 AM Working Time

Figure 9: Data after the encryption using the proposed RC4-EA encryption method

recovering the secret key. In the proposed RC4-EA encryption method, Evolutionary Algorithm (EA) is used for generating a dynamic random secret key as a seed for RC4 in KSA algorithm to create a pseudo-random initial state. Then, PRGA generates a pseudo-random output final key stream. This final key stream will XORed with the plaintext to generate the ciphertext.

The main advantage of the proposed RC4-EA method is increasing the security of the system by generating a dynamic random secret key. Hence, overcomes the drawback of a non-random secret key as a seed used in RC4. Which makes it difficult for the hacker to trace the plaintext and the secret key used to generate the final key stream. Several experiments with different secret key length (128 bits and 256 bits), were conducted to evaluate the proposed RC4-EA method. Experiment's results show that the proposed RC4-EA encryption method enhances the encryption times and the throughputs compared to the original RC4 algorithm. The proposed RC4-EA encryption method is applied for ciphering the data in Content Management System, to keep these data in high confidential authentication.

# References

[1] Anonymous, *RC4 Source Code*, CypherPunks mailing list, Sept. 1994. (http://cypherpunks.venona.com/date/1994/09/msg00304.html)

[2] T. Back, "Parallel optimisation of evolutionary algorithms", in *International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem*, Israel, Springer, Berlin, vol. 866, pp. 418-427, 1994.

[3] T. Back, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, NewYork, 1st Edition, 1996.

[4] T. Back, H. P. Schwefel, "An overview of evolutionary algorithms for parameter optimization", *Evolutionary Computation*, vol. 1, n0000o. 1, pp. 1-23, 1993.

[5] D. S. Abd Elminaam, H. M. Abdual Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms", *International Journal of Network Security*, vol. 10, no. 3, pp. 213-219, 2010.

[6] S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the key scheduling algorithm of RC4", in *Proceedings of Annual Workshop on Selected Areas in Cryptography*, Springer, Toronto, vol. 2259, pp. 1-24, 2001.

[7] A. Grosul, D. Wallach, "A related-key cryptanalysis of RC4", Department of computer science, Rice University, Technical report, pp. 01-358, June 2000.

[8] S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B. Sinha, "High-performance hardware implementation for RC4 stream cipher", *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 730-743, 2013.

[9] M. M. Hammood, K. Yoshigoe and A. M. Sagheer, "RC4-2S: RC4 stream cipher with two state tables", *Information Technology Convergence*, Lecture Notes in Electrical Engineering, Springer Science - Business Media Dordrecht, vol. 253, pp. 13-20, 2013.

[10] Ch. Lin, Y. Li, K. Lv and C. C. Chang, "Ciphertext-auditable identity-based encryption", *International Journal of Network Security*, vol. 17, no. 1, pp. 23-28, 2015.

[11] N. Lourenco, F. Pereira, and E. Costa, "Evolving evolutionary algorithms", in *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO'12)*, ACM New York, NY, USA, pp. 51-58, 2012.

[12] J. Lv, B. Zhang and D. Lin, "Distinguishing attacks on RC4 and a new improvement of the cipher", *International Association for Cryptologic Research (IACR)*, 2013. (https://eprint.iacr.org/2013/176.pdf)

[13] I. Mantin, A. Shamir, "A practical attack on broadcast RC4", *FastSoftware Encryption*, LNCS 2355, pp. 152-164, 2001.

[14] Ch. N. Mathur and K. P. Subbalakshmi, "A light weight enhancement to RC4 based security for resource constrained wireless devices", *International Journal of Network Security*, vol. 5, no. 2, pp. 205-212, 2007.

[15] A. Mousa, and A. Hamad, "Evaluation of the RC4 algorithm for data encryption", *International Jornal of Computer Science and Applications*, vol. 3, no. 2, pp. 44-56, 2006.

[16] Pardeep and Pushpendra, "A pragmatic study over the different stream cipher and on different flavor of RC4 stream cipher", *International Journal of Computer Science and Network Security*, vol. 12, no. 3, pp. 37-42, 2012.

[17] G. Paul, S. Maitra, "RC4 state in formation at any stage reveals the secret key", in *Presented in the 14th Annual Workshop on Selected Areas in Cryptography*, SAC, Ottawa, Canada, LNCS vol. 4876, pp. 360-377, 2007.

[18] S. Paul, and B. Preneel, "A new weakness in the RC4 keystream generator", *Fast Software Encryption (FSE'04)*, Springer-Verlag, vol. 3017, pp. 245-259, 2004.

[19] K. Sharma, M. K. Ghose, D. Kumar, R. Singh, and V. Pandey, "A comparative study of various security approaches used in wireless sensor networks", *International Journal of Advanced Science and Technology*, vol. 17, pp. 31-44, Apr. 2010.

[20] S. O. Sharif, S. P. Mansoor, "Performance analysis of stream cipher algorithms", in *3rd International Conference on Advanced Computer Theory and Engineering (ICATE'10)*, vol. 1, pp. 522-525, 2010.

[21] A. K. Singha, S. G. Samaddar, S. R. Sahooc, and G. Mathewd, "Increasing robustness of RC4 family for automated selection of ciphersuites", in *International Conference on Communication Technology and System Design (ICCTSD'12)*, vol. 30, pp. 4552, 2012.

[22] L. Stosic, and M. Bogdanovic, "RC4 stream cipher and possible attacks on WEP", *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 3, pp. 110-114, 2012.

[23] P. Subsorn, S. Limwiriyakul, "A Comparative analysis of the security of internet banking in Australia: A customer perspective", in *2nd International Cyber Resilience Conference*, Western Australia, pp. 70-83, 2011.

[24] P. Subsorn, S. Limwiriyakul, "A comparative analysis of internet banking security in Thailand: A customer perspective", in Proceedings of Engineering, vol. 32, no. 37, pp. 260272, 2012.

[25] Q. Yu, C. Zhang, "RC4 state and its applications", in *Ninth Annual International Conference on Privacy, Security and Trust*, pp. 264-269, 2011.

[26] X. Yu, and M. Gen, *Introduction to Evolutionary Algorithms*, Springer London Dordrecht Heidelberg New York, ISBN: 978-1-84996-129-5, XVII, pp. 418, 2010.

**Ashraf Aboshosha** graduated with a B.Sc. in industrial electronics from Menoufia University, Egypt at 1990. At 1997 he received his M.Sc. in automatic control and measurement engineering. From 1997 to 1998 he was guest researcher at research centre Jlich (FZJ), Germany. From 2000 to 2004 he was a doctoral student (DAAD-scholarship) at Eberhard-Karls-University, Tbingen, Germany. Where he received his Doctoral degree (Dr. rer. nat.) at 2004. He is the CEO of ICGST LLC, Delaware, USA.

**Kamal Abdelraouf ElDahshan** is a professor of Computer Science and Information Systems at Al-Azhar University in Cairo, Egypt. An Egyptian national and graduate of Cairo University, he obtained his doctoral degree from the Universit de Technologie de Compigne in France, where he also taught for several years. During his extended stay in France, he also worked at the prestigious Institute National de Tlcommunications in Paris. Professor ElDahshan's extensive international research, teaching, and consulting experiences have spanned four continents and include academic institutions as well as government and private organizations. He taught at Virginia Tech as a visiting professor; he was a Consultant to the Egyptian Cabinet Information and Decision Support Centre (IDSC); and he was a senior advisor to the Ministry of Education and Deputy Director of the National Technology Development Centre. Prof. ElDahshan has taught graduate and undergraduate courses in information resources and centers, information systems, systems analysis and design, and expert systems. Professor ElDahshan is a professional Fellow on Open Educational Resources as recognized by the United States Department of State. Prof. Eldahshan wants to work in collaboration with the Ministry of Education to develop educational material for K-12 levels. Prof. Eldahshan is interested in training instructors to be able to use OER in their teaching and hopes to make his university a center of excellence in OER and offer services to other universities in the country.

**Eman K. Elsayed** Bachelor of Science from computer science Department, Cairo University 1994, Master of computer science from Cairo university 1999, and computer science PHD 2005 from Alazhar university. I Published eleven papers until 2010 in data mining, ontology and e-learning. I am a member in egyptian mathematical society and inteligent computer and information systems society.

**Ahmed A. Elngar** graduated with a B.Sc. in computer Science from computer science Department, Al-Azhar University 2004, Master of computer science in Intrusion Detection System (IDS) from Ain Shanm university 2012. Now he is a P.hD student at computer science Department, Al-Azhar University. Also he is a member in Egyptian Mathematical Society (EMS) and International Rough Set Society(IRSS).

# Improved RFID Authentication Protocol Based on Randomized McEliece Cryptosystem

Noureddine Chikouche[1], Foudil Cherif[2], Pierre-Louis Cayrel[3], and Mohamed Benmohammed[4]
*(Corresponding author: Noureddine Chikouche)*

Computer Science Department, Mohamed Boudiaf University of M'sila[1]
BP 166 ichebilia, 28000 M'sila, Algeria
(Email: chiknour28@univ-msila.dz)
Computer Science Department, LESIA Laboratory, Mohamed Khider University of Biskra[2]
BP 145 RP, 07000 Biskra, Algeria
Laboratoire Hubert Curien, UMR CNRS 5516[3]
Bâtiment F18 rue du professeur Benoît Lauras, 42000 Saint-Etienne, France
LIRE Laboratory, University of Constantine 2[4]
P.O. Box 325, City Ain El Bey 25017 Constantine, Algeria

*(Received Dec. 15, 2014; revised and accepted Apr. 2 & Apr. 17, 2015)*

## Abstract

Among the embedded systems which were quickly developed during the last years and that were used in various domains (e.g. access control, health, ...) we can cite radio frequency identification (RFID). In this paper, we propose an improved mutual authentication protocol in RFID systems based on the randomized McEliece cryptosystem. The McEliece cryptosystem is not only very fast, but it is resistant to quantum computing and it does not require any crypto-processor. Our work includes a comparison between the improved protocol and different existing protocols based on error-correcting codes in terms of security and performance. Security and privacy properties are proved, and the performance of the proposed authentication protocol is analysed in terms of storage requirement, communication cost and computational cost.

*Keywords: Authentication protocol, McEliece cryptosystem, RFID*

## 1 Introduction

Among the embedded systems which were quickly developed during the last years and that were used in various domains (e.g. access control, supply chain management,health, ...) we can cite radio frequency identification (RFID). The typical RFID system consists of three entities: tags, readers and server. The tag is a small electronic chip supplemented with an antenna that can transmit and receive data, the reader i.e. a device to communicate with tags by radio waves. The server (or back-end) is a centralized place that hosts all data regarding access permissions and may be consulted by the reader. The use of cryptographic primitives in low-cost RFID tags is limited because the space memory available is restricted, and the computational capabilities are limited. The lowest cost RFID tags are assumed to have the capability of performing bitwise operations (e.g. xor, and, ...), bit shifts (e.g. rotate, logical shift, ...) and random number generator.

The code-based cryptography is a very important research area and it is applied in different schemes. Its advantages are: high-speed encryption and decryption compared to public-key cryptosystems based on number theory. It does not require a crypto-processor and based on difficult problems NP-complete (syndrome decoding, ...). It resists to quantum attacks, and it uses different schemes, such as: public-key cryptosystems, identification schemes, secret sharing and signature [31].

The major problem was the size of public key. Recently, code-based cryptosystems were presented with small key sizes, for example, we quote [3, 22]. In the majority of RFID authentication protocols, the tag does not require a generator matrix or other matrices, but it stores the codeword with the necessary information. RFID authentication protocols based on error-correcting codes use various schemes: error-correcting code with secret parameters [8, 9, 26], randomized Niederreiter cryptosystem [11, 30], Quasi-Dyadic Fix Domain Shrinking [28] and randomized McEliece cryptosystem [19].

In order to have secure authentication protocols, it is important that a RFID authentication protocol own security and privacy properties:

**Secrecy.** It provides that the identifier of the tag or secret data is never send in clear to air on the interface radio frequency which can be spied.

**Mutual Authentication.** A RFID authentication protocol achieves mutual authentication that is to say; it achieves the tag's authentication and the reader's authentication. In tag's authentication, the reader has to be capable of verifying a correct tag to authenticate and to identify a tag in complete safety. In reader's authentication, a tag has to be able to confirm that it communicates with the legitimate reader.

**Untraceability.** the untraceability is one of the privacy properties. The tag is untraceable if an intruder cannot tell whether he has seen the same tag twice or two different tags [12].

**Desynchronization Resilience.** This property specifies for RFID protocols updating a shared secret before terminating the protocol. The definition of this property is as follows: in session $(i)$, the intruder can block or modify the exchanged messages between the reader and the tag. In the next session, the authentication process is will fail because the tag and the reader are not correlated.

**Forward Secrecy.** One of the abilities of the intruder is to compromise secrets stored in the tag. The property of forward secrecy signifies to protect the previous communications from a tag even assuming the tag has been compromised.

**Resist Replay Attack.** The intruder can listen to the message answer of the tag and to the reader. It will broadcast the message listened without modification to the reader later.

We propose in this paper an improved RFID mutual authentication protocol using code-based scheme. Our protocol based on randomized McEliece cryptosystem, uses an efficient decoding/encoding algorithm to generate an error vector of fixed weight. The only datum stored in tag is a dynamic identifier, and it is updated before the end of the session and without the need to do exhaustive search to obtain the identifier from a database. The paper includes a comparison between the new protocol and different protocols based on error-correcting codes in terms of security and performance. Our protocol proves security and privacy properties. Using the AVISPA (Automated Validation of Internet Security Protocols and Applications) tools [1], we prove the security requirements. We use the privacy model of Ouafi and Phan [25] to verify the untraceability property. The performance of the proposed authentication protocol is analysed in terms of storage requirements, communication cost and computational cost.

The rest of this paper is structured as follows: Section 2 presents the basic concepts of code-based cryptography. Section 3 presents related work. We describe our proposed protocol in Section 4. In Section 5, we prove the security and privacy requirements. Section 6 presents the comparative study in terms of performance. Finally, the paper ends with a general conclusion.

# 2 Code-based Cryptography

$\mathcal{C}[n,k,d]$ is a binary linear code, where $n$ is length and $k$ is dimension which stands a generator matrix $\mathcal{G}'$ ($k$ and $n$ are positive integers and $k < n$). The minimum distance $d$ is the smallest weight of any non-zero codeword in the code. The codeword $c$ of $n$ bits is $m\mathcal{G}$, where $m$ is binary string with length $k$ and $\mathcal{G}$ is a public-key matrix. The encoded codeword is $c' = c \oplus e$, where $e$ is an error vector of length $n$ and weight $t = \mathsf{wt}(e)$, with $t$ is less than or equal to $\lfloor \frac{d-1}{2} \rfloor$.

## 2.1 McEliece Cryptosystem

The McEliece cryptosystem [20] is the first public key cryptosystem using algebraic coding theory and based on the problem of computational dual decoding syndrome. The idea of McEliece is to hide the corresponding codeword to the message by adding as an error vector while still being able to correct them. If the correction method is kept secret, then only the recipient will be able to recover the original message. We describe this cryptosystem as follows.

**Key Generation Algorithm**

- choose $n, k$ and $d$
- randomly generate a generator matrix $\mathcal{G}'$ of an $[n,k,d]$ binary Goppa code $\mathcal{C}$,
- randomly generate a $n \times n$ binary permutation matrix $P$,
- randomly generate a $k \times k$ binary invertible matrix $S'$,
- compute $\mathcal{G} = S'\mathcal{G}'P$,
- public key is $(\mathcal{G}, t)$, where $t$ integer $< \dfrac{d}{2}$,
- private key is $(S', \mathcal{G}', P, \mathcal{A}(.))$, where $\mathcal{A}(.)$ is a polynomial-time decoding algorithm until $< \dfrac{d}{2}$ errors (like for instance the Patterson algorithm for binary Goppa codes).

**Encryption Algorithm**

- $m$ message with length $k$,
- randomly generate $e$ of weight $t$,
- output $c' = m\mathcal{G} \oplus e$, where $\mathsf{wt}(e) = t$.

**Decryption Algorithm**

- compute $z = c'P^{-1}$,
- compute $y = \mathcal{A}(z)$,
- output $m = yS'^{-1}$.

### 2.2 Randomized McEliece Cryptosystem

Nojima et al. [24] prove that padding the plaintext with a random bit-string provides the semantic security against chosen plaintext attack (IND-CPA) for the McEliece cryptosystem with the standard assumptions.

The standard assumptions are: the syndrome decoding (SD) problem is hard and the public-key is indistinguishable.

The randomized McEliece is a probabilistic cryptosystem, whose encryption algorithm of message is as follows:

$$c' = c \oplus e = [r \parallel m]\, \mathcal{G} \oplus e = (r\mathcal{G}_1 \oplus e) \oplus m\mathcal{G}_2 \qquad (1)$$

where:

- $\mathcal{G} = \left[\begin{smallmatrix}\mathcal{G}_1 \\ \mathcal{G}_2\end{smallmatrix}\right]$
- $k_1$ and $k_2$: two integers such that $k = k_1 + k_2$ and $k_1 < bk$ where $b < 1$ (e.g. $b = \frac{9}{10}$ [24]),
- $\mathcal{G}_1$ and $\mathcal{G}_2$ : matrices with $k_1 \times n$ and $k_2 \times n$, respectively,
- $r$: random string with length $k_1$,
- $m$: message with length $k_2$.

The encryption algorithm encrypts $[r\|m]$ instead of $m$ itself. The decryption algorithm is almost the same as original McEliece, the difference is that it outputs only the last $k_2$ bits of the decrypted string.

### 2.3 Encoding Constant Weight Words

To transform a binary string into error vector (bijective) or encode/decode constant weight words, we have two methods: the enumerative method [10, 27] and the recursive method [29]. We are interested in the enumerative method, which is based on the following bijective application:

$$\begin{array}{llll} \phi_{n,t}: & \left[0, \binom{n}{t}\right[ & \longrightarrow & \mathcal{W}_{n,t} := \{x \in \mathbb{F}_q^n | \mathsf{wt}(x) = t\} \\ & x & \mapsto & (i_1, \cdots, i_t) \end{array}$$

$\mathcal{W}_{n,t}$ is represented by its non-zero positions in increasing order $0 \leq i_1 < i_2 < \cdots < i_t \leq n-1$ and length of $x$ is $\ell = \left\lfloor \log_2 \binom{n}{t} \right\rfloor$.

The inverse application is defined as follows:

$$\begin{array}{llll} \phi_{n,t}^{-1}: & \mathcal{W}_{n,t} & \longrightarrow & \left[0, \binom{n}{t}\right[ \\ & (i_1, \cdots, i_t) & \mapsto & \binom{i_1}{1} + \binom{i_2}{2} + \ldots + \binom{i_t}{t} \end{array}$$

The cost of a bijective application is $\mathcal{O}(t\ell^2)$ binary operations. The decoding algorithm $\phi_{n,t}$ is proposed by [10, 27] as follows (Algorithm 1).

**Algorithm 1** Enumerative decoding

```
Data $x \in \left[0, \binom{n}{t}\right[$
Result $t$ integers $0 \leq i_1 < i_2 < \cdots < i_t \leq n-1$
$j \leftarrow t$
while $j > 0$ do
    $i_j \leftarrow$ invert-binomial $(x, j)$
    $x \leftarrow x - \binom{i_j}{j}$
    $j \leftarrow j - 1$
end while
where invert-binomial $(x, j)$ returns the integer $i$ such
  that $\binom{i}{j} \leq x < \binom{i+1}{j}$
```

# 3 Related Work

In a survey of design and implementation of authentication protocols on RFID systems, we can find many protocols developed using various algebraic and cryptographic primitives (asymmetric cryptosystems, symmetric cryptosystems, hash function, bitwise operators, ...), such as [5, 7, 17, 23, 32, 33, 34, 35]. Our work is articulated on recent RFID authentication protocols that use error-correcting codes.

Park [26] proposed a one-way authentication protocol to provide untraceability which is based on the secret-key certificate and the algebraic structure of the error-correcting code. This protocol is designed for wireless mobile communication systems. We study this protocol because the computational capabilities of Mobile subscriber is limited like RFID tag. This protocol does not achieve untraceability because the weight of $e$ in session $(i)$ is the same weight as in session $(j)$ with equal $t$. If the intruder knows $d$ or $t$, so the intruder can trace of the legitimate tag. Also, this protocol does not resist desynchronization attacks because the tag and the reader store a number of the last session and do not use a secret synchronization value.

In [11], authors proposed an authentication protocol based on the randomized Niederreiter cryptosystem and the amelioration of the protocol [30]. This protocol does not achieve forward secrecy because the data stored in tag is static and does not achieve the reader's authentication.

Chien and Laih [9] proposed a RFID authentication protocol based on error-correcting codes with secret parameters. This protocol uses a confusion scheme to avoid traceability attacks. The data stored in tag is static, therefore, this protocol does not achieve forward secrecy.

Sekino et al. [28] proposed a challenge-response authentication protocol based on Quasi-Dyadic Fix Domain Shrinking that combines Niederreiter personalized public-key cryptosystem (P$^2$KC) [18] with Quasi-dyadic (Goppa) codes [22]. The authors reduce the size of the public-key matrix stored in tag of protocol [11], but it remains relatively important compared to the resources of low-cost tag. Also, the information stored in tag is static, therefore, this protocol does not achieve forward secrecy.

Malek and Miri [19] proposed a RFID authentication protocol based on randomized McEliece public-key cryptosystem. In this protocol, the tag can communicate with a set of authorised readers. This protocol achieves the untraceability property because the identifier is modified in each session. Concerning the desynchronization attack, if the intruder modifies a last message, then the identifier stored in reader is different to identifier stored in tag. Thus, this protocol does not resist the desynchronisation attacks. In other hand, in the phase of reader's authentication, the tag computes and uses the circulant matrix, this requires a more complex computation and important space in volatile memory.

# 4 Our Improved Protocol

In this section, we propose an improved mutual protocol based on randomized McEliece cryptosystem. To better describe our proposed protocol, we use the notations given in Section 2 and Table 1.

Table 1: Notations

| | |
|---|---|
| $T, R, S$ | The tag, the reader and the server |
| $N_R$ | Random number generated by $R$ |
| $g(.)$ | Pseudo-random function |
| $\|\|$ | Concatenation of two inputs |
| $t, t'$ | Integer numbers |
| $x$ | Random number, with $x \in \big[0, \binom{n}{t'}\big[$ |
| $\phi_{n,t'}(x)$ | decoding bijective application (transform $x$ into error vector $e$) |
| $e$ | Error vector of length $n$ and weight $t' < t$ where $t = \lfloor(d-1)\rfloor/2$ |
| $id$ | Identifier of tag, with binary length $k_2$ |
| $r, r'$ | Random numbers with binary length $k_1$ |
| $c_r$ | Codeword, where $c_r = r\mathcal{G}_1$ |
| $c_{r'}$ | Codeword, where $c_{r'} = r'\mathcal{G}_1$ |
| $c_{id}$ | Codeword, where $c_{id} = id\mathcal{G}_2$ |
| $DID$ | Dynamic ID, codeword with length $n$, where $DID = c_r \oplus c_{id}$ |
| $c_{r_{old}}$,$c_{r_{new}}$ | Two secret synchronization codewords, where $c_{r_{old}} = r_{old}\mathcal{G}_1$ and $c_{r_{new}} = r_{new}\mathcal{G}_1$ |

## 4.1 System Model

The RFID system consists of three entities: tag $T$, reader $R$ and server $S$.

- The tag $T$ is low-cost and passive. It stores the dynamic identity ($DID$) which is strictly confidential. $T$ implements an application $\phi_{n,t'}$ and pseudo-random numbers generator (PRNG) to generate $x$ and compute $g(.)$. It also supports bitwise operations (xor, and, ...). A tag has a rewritable memory that may not be tamper-resistant.

- The reader $R$ can generate pseudo-random numbers.

- The server $S$ has the sufficient storage space and computational resources. We implement algorithms of $\phi^{-1}_{n,t'}$ and PRNG. Server $S$ can decode the message received from $T$, then, we implement encryption/decryption of randomized McEliece cryptosystem with public-key matrix $\mathcal{G}$, private-key matrices and a polynomial-time decoding algorithm $\mathcal{A}(.)$. The server contains the database which includes $\{id, c_{id}, c_{r_{old}}, c_{r_{new}}\}$.

In our work, we propose to use $\phi_{n,t'}(x)$ as follows (Algorithm 2).

**Algorithm 2** Generation a error vector

```
Randomly choose $x \in \big[0, \binom{n}{t}\big[$
repeat
    determine the largest $t'$ such that $x \in \big[0, \binom{n}{t'}\big[$
until $t' < t$
compute $\phi_{n,t'}(x) = e$ where $\mathsf{wt}(e) = t' < t$
```

We will choose $t'$ such that the syndrome decoding problem (most efficient algorithm) remains hard.

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 in the authentication phase since it makes it open to attacks on the authentication protocol.

## 4.2 Description of Our Proposed Protocol

The proposed Protocol is divided into two phases: the initialization phase and the mutual authentication phase.

### 4.2.1 Initialization Phase

The server generates a random binary Goppa code $\mathcal{C}[n, k, d]$ as specified by the generator matrix $\mathcal{G}'$, where $\mathcal{G} = S'\mathcal{G}'P$ and $\mathcal{G}$ is public-key. The server $S$ generates random values using PRNG, $id$ the unique identifier of tag and the random number $r$. It computes $c_r = r\mathcal{G}_1$, $c_{id} = id\mathcal{G}_2$ and $DID = c_r \oplus c_{id}$, and initializes $c_{r_{old}}$ and $c_{r_{new}}$ by $c_r$. Then, the server (registration center) sends $DID$ to the tag through a secure channel, where $DID$ is strictly confidential. $S$ stored in the database $\{id, c_{id}, c_{r_{old}}, c_{r_{new}}\}$ for each tag.

### 4.2.2 Mutual Authentication Phase

The mutual authentication phase is described as follows (and in Figure 1).

**Step 1.** ***Tag's Authentication***

**Step 1.1.** $R$ generates a nonce $N_R$ and sends it as a request to the tag $T$.

**Step 1.2.** $T$ generates a random number $x \in \big[0, \log_2\binom{n}{t'}\big[$ and $t' \in [1, t[$, and computes error vector $e$ with $\mathsf{wt}(e) = t'$ from $\phi_{n,t'}(x)$, $c' = DID \oplus e$ and $P = g(N_R \parallel x \parallel DID)$.

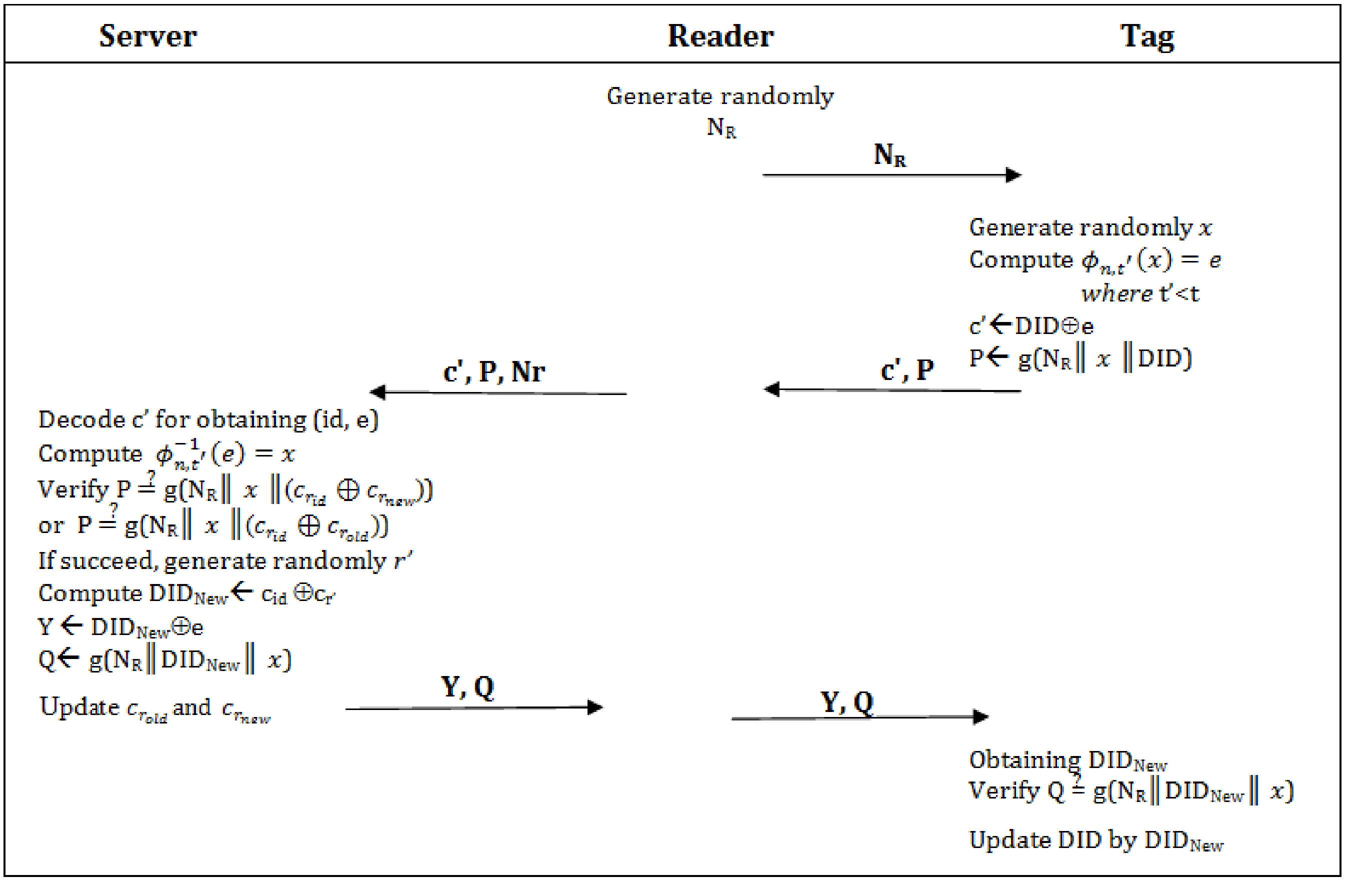

Server
Reader
Tag
Generate randomly N_R
N_R
Generate randomly x
Compute φ_{n,t'}(x) = e where t'<t
c'←DID⊕e
P← g(N_R‖ x ‖DID)
c', P
c', P, Nr
Decode c' for obtaining (id, e)
Compute φ^{-1}_{n,t'}(e) = x
Verify P =? g(N_R‖ x ‖(c_{r_id} ⊕ c_{r_new}))
or P =? g(N_R‖ x ‖(c_{r_id} ⊕ c_{r_old}))
If succeed, generate randomly r'
Compute DID_New← c_id ⊕c_r'
Y ← DID_New⊕e
Q← g(N_R‖DID_New‖ x)
Update c_{r_old} and c_{r_new}
Y, Q
Y, Q
Obtaining DID_New
Verify Q =? g(N_R‖DID_New‖ x)
Update DID by DID_New

Figure 1: Our proposed protocol

**Step 1.3.** $T$ sends $c'$ with $P$ to the reader, and resends the received $c'$, message $P$ and nonce $N_R$ to the server $S$.

**Step 1.4.** $S$ performs a decoding algorithm $\mathcal{A}(.)$ with private key matrices and identifies the error vector $e$ as well as $id$ and $r$. From $id$, in database, the server retrieves the values of $c_{id}, c_{r_{old}}, c_{r_{new}}$ and computes $x = \phi^{-1}_{n,t'}(e)$ and $P_1 = g(N_R \parallel x \parallel (c_{id} \oplus c_r))$ (either $c_{r_{old}}$ or $c_{r_{new}}$). $S$ verifies if $P_1 \overset{?}{=} P$, if they are equal, the tag's authentication is successful; otherwise the tag's authentication has failed.

**Step 2.** ***Reader's Authentication***

**Step 2.1.** In this case the tag's authentication is successful. The server generates a random number $r'$ and computes $c_{r'} = r'\mathcal{G}_1$ and $DID_{New} = c_{id} \oplus c_{r'}$. It computes $Y = DID_{New} \oplus e$ and $Q = g(N_R \parallel DID_{New} \parallel x)$. It updates $c_{r_{old}} \leftarrow c_{r_{new}}$ and $c_{r_{new}} \leftarrow c_{r'}$, only in case the matched $c_r$ is $c_{r_{new}}$.

**Step 2.2.** $S$ sends $Y$ and $Q$ to the reader and resends the received message to $T$.

**Step 2.3.** $T$ obtains $DID_{New}$ by computing $Y \oplus e$ and calculates $Q_1 = g(N_R \parallel DID_{New} \parallel x)$. $T$ verifies if $Q_1 \overset{?}{=} Q$, if they are equal, the reader's authentication is successful; otherwise the authentication of the reader will fail.

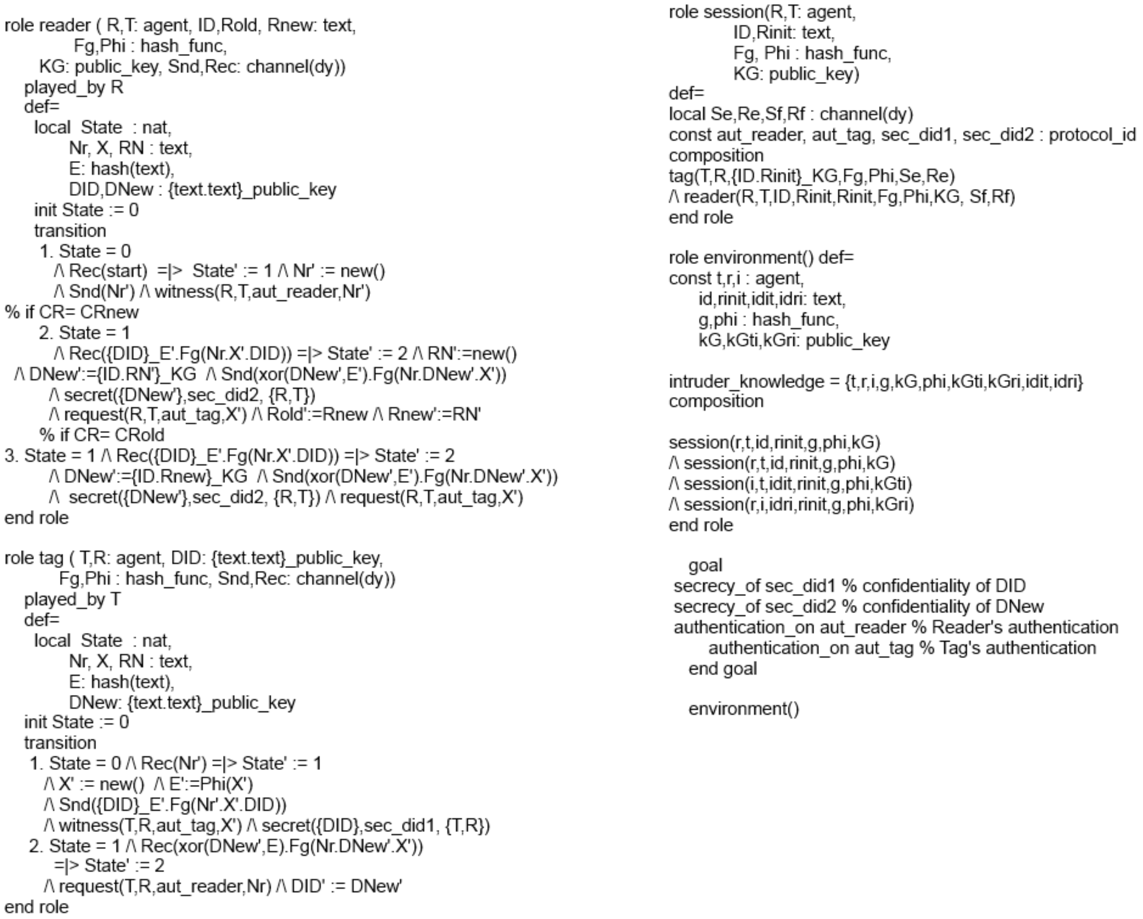
**Step 2.4.** $T$ updates the dynamic identifier by the value of $DID_{New}$, if reader's authentication is successful.

## 5 Security and Privacy Analysis

A secure RFID authentication protocol should provide mutual authentication, secrecy, untraceability, desynchronization resilience, forward secrecy and replay attack resisting. In this section, we discuss the security and privacy requirements of proposed protocol and others protocols. Table 2 presents the security comparison between the existing protocols and the proposed protocol.

### 5.1 Automated Verification

We choose AVISPA tools (Automated Validation of Internet Security Protocols and Applications) [1] to verify the security properties for the following reasons: the tools uses various techniques of validation (Model-checking, automate trees, Solver SAT and resolution of constraints). The AVISPA platform is the analyzer which models a



```
role reader ( R,T: agent, ID,Rold, Rnew: text,
            Fg,Phi : hash_func,
      KG: public_key, Snd,Rec: channel(dy))
    played_by R
    def=
      local  State  : nat,
            Nr, X, RN : text,
            E: hash(text),
            DID,DNew : {text.text}_public_key
      init State := 0
      transition
       1. State = 0
         /\ Rec(start)  =|>  State' := 1 /\ Nr' := new()
         /\ Snd(Nr') /\ witness(R,T,aut_reader,Nr')
% if CR= CRnew
       2. State = 1
         /\ Rec({DID}_E'.Fg(Nr.X'.DID)) =|> State' := 2 /\ RN':=new()
  /\ DNew':={ID.RN'}_KG  /\ Snd(xor(DNew',E').Fg(Nr.DNew'.X'))
        /\ secret({DNew'},sec_did2, {R,T})
        /\ request(R,T,aut_tag,X') /\ Rold':=Rnew /\ Rnew':=RN'
       % if CR= CRold
3. State = 1 /\ Rec({DID}_E'.Fg(Nr.X'.DID)) =|> State' := 2
        /\ DNew':={ID.Rnew}_KG  /\ Snd(xor(DNew',E').Fg(Nr.DNew'.X'))
        /\  secret({DNew'},sec_did2, {R,T}) /\ request(R,T,aut_tag,X')
end role

role tag ( T,R: agent, DID: {text.text}_public_key,
          Fg,Phi : hash_func, Snd,Rec: channel(dy))
    played_by T
    def=
      local  State  : nat,
            Nr, X, RN : text,
            E: hash(text),
            DNew: {text.text}_public_key
    init State := 0
    transition
     1. State = 0 /\ Rec(Nr') =|> State' := 1
       /\ X' := new()  /\ E':=Phi(X')
       /\ Snd({DID}_E'.Fg(Nr'.X'.DID))
       /\ witness(T,R,aut_tag,X') /\ secret({DID},sec_did1, {T,R})
     2. State = 1 /\ Rec(xor(DNew',E).Fg(Nr.DNew'.X'))
         =|> State' := 2
       /\ request(T,R,aut_reader,Nr) /\ DID' := DNew'
end role
```

```
role session(R,T: agent,
            ID,Rinit: text,
            Fg, Phi : hash_func,
            KG: public_key)
def=
local Se,Re,Sf,Rf : channel(dy)
const aut_reader, aut_tag, sec_did1, sec_did2 : protocol_id
composition
tag(T,R,{ID.Rinit}_KG,Fg,Phi,Se,Re)
/\ reader(R,T,ID,Rinit,Rinit,Fg,Phi,KG, Sf,Rf)
end role

role environment() def=
const t,r,i : agent,
     id,rinit,idit,idri: text,
     g,phi : hash_func,
     kG,kGti,kGri: public_key

intruder_knowledge = {t,r,i,g,kG,phi,kGti,kGri,idit,idri}
composition

session(r,t,id,rinit,g,phi,kG)
/\ session(r,t,id,rinit,g,phi,kG)
/\ session(i,t,idit,rinit,g,phi,kGti)
/\ session(r,i,idri,rinit,g,phi,kGri)
end role

   goal
 secrecy_of sec_did1 % confidentiality of DID
 secrecy_of sec_did2 % confidentiality of DNew
 authentication_on aut_reader % Reader's authentication
      authentication_on aut_tag % Tag's authentication
   end goal

   environment()
```

Figure 2: Specification of our protocol by HLPSL

Table 2: Comparison of security and privacy properties

| | M.A | D.C | Unt | D.R | F.S | R.R |
|---|---|---|---|---|---|---|
| Park [26] | N | Y | N | N | Y | Y |
| Cui et al. [11] | N | Y | Y | Y | N | Y |
| Chien-Laih [9] | Y | Y | Y | Y | N | Y |
| Sekino et al. [28] | N | Y | Y | Y | N | Y |
| Malek-Miri [19] | Y | Y | Y | N | Y | Y |
| Our Protocol | Y | Y | Y | Y | Y | Y |

M.A: Mutual Authentication, D.C: Data Confidentiality
Unt: Untraceability, D.R: Desynchronization resilience
F.S: Forward secrecy, R.R: Resist replay attacks

big number of cryptographic protocols. These tools can detect passive and active attacks, like replay and man-in-the-middle attacks. AVISPA tools are based on only one specification language named HLPSL language (High-Level Protocol Specification Language) [2].

HLPSL is a formal, expressive, modular and 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. HLPSL can specify the secrecy and the authentication properties.

The intruder model agreed in HLPSL is Dolev-Yao model [13]. 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 an intruder can decrypt a message $m$ encrypted with key $k$ if it has the opposite of that key. The second hypothesis which is *the intruder is the network* means that, the intruder has complete control over the channel of communication between the reader and the tag. It can intercept any message passing through the network, block or modify messages and it can also derive new messages from its initial knowledge.

Our protocol requires the primitives: PRNG, nonce xor-operator and McEliece cryptosystem. The randomized McEliece cryptosystem requires the primitives: public key, private key, application $\phi_{n,t'}(.)$ and the decoding algorithm $\mathcal{A}(.)$ which is used with a private key to obtain $id$ and $e$. The application $\phi_{n,t'}(.)$ is bijective, but the intruder cannot find $x$ without knowing the value of $t'$, and the result of this application $e$ does not circulate clearly in the channel, then we can model it by a hash function $Phi(x)$. The intruder will know this function, therefore he will be able to compute the error vector but not invert values of $Phi^{-1}(x)$ (unless he already knows $x$).

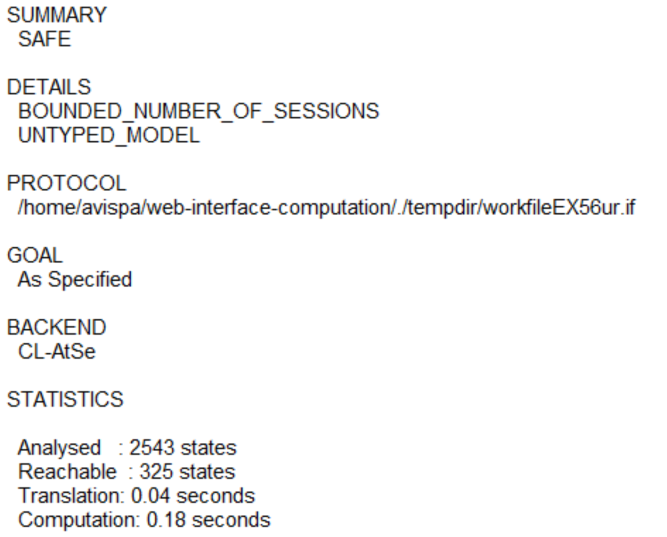
Concerning the message $DID \oplus e$, we cannot specify it in HLPSL by $xor(DID, E)$ because the reader does not use the algebraic properties of or-exclusive operator (e.g. neutral element) to obtain $id$ and $e$. To retrieve these values, we apply the private decoding algorithm $\mathcal{A}(.)$ and the private key of McEliece. $DID \oplus e$ means the encoding $DID$ by $e$, where $DID$ is encryption of $[r \parallel id]$ by public key $\mathcal{G}$. The reader (server) obtaining the value $DID$ and $e$ uses the private decoding algorithm $\mathcal{A}(.)$. Therefore, we propose to specify this message in HLPSL by $\{DID\}\_E$. In the other hand, we can specify the message $DID_{New} \oplus e$ by $xor(DNew, E)$ (last message from reader to tag) because the objective of the tag is to retrieve the value of $DID_{New}$ using the algebraic properties of xor operator.

The Figure 2 shows the specification of our protocol by HLPSL. In our protocol, the honest participants are the reader $R$ and the tag $T$. Then, we have two basic roles, the tag and the reader. We can define a session role which all the basic roles are instanced with concrete arguments. In the *tag*, we initialise the argument $DID$ by {ID.Rinit}_kG. In the *reader*, we initialise the values $Rold$ and $Rnew$ by $Rinit$. We provide a validation of properties: the tag's authentication (*aut_tag*), the reader's authentication (*aut_reader*), the secrecy of current $DID$ (*sec_did*1), and the secrecy of the new $DID$ (*sec_did*2).

The result of verification of our protocol by AVISPA tools is presented in Figure 3. This result clearly means that there is no attack detected (replay or man-in-the-middle attacks). We can thus deduct that the diagnostic of AVISPA tools for our protocol is secure.

## 5.2 Privacy Verification

In the literature of formal verification of privacy properties, we can find many privacy models. The privacy model proposed by Juels and Weis [16] is based on the notion of indistinguishability. Ouafi and Phan model [25] is based on the Juels-Weis model. Authors added several definitions in the untraceability property.



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 UNTYPED_MODEL

PROTOCOL
 /home/avispa/web-interface-computation/./tempdir/workfileEX56ur.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 2543 states
 Reachable  : 325 states
 Translation: 0.04 seconds
 Computation: 0.18 seconds
```

Figure 3: The result of the verification using CL-AtSe tool of our protocol

In Ouafi and Phan model, a protocol party is a tag $T \in Tags$ or a reader $R \in Readers$ interacting in protocol sessions as per the protocol specifications until the end of the session. The adversary is allowed to run the following queries:

- Execute $(R, T, i)$ query. This query models the passive attacks. The adversary $A$ eavesdrops the communication channel between $T$ and $R$ and gets reading access to the exchanged messages in session $i$ of a truthful protocol execution.
- Send $(U, V, m, i)$ query. This query models active attacks by allowing the adversary $A$ to impersonate some reader $U \in Readers$ (respectively tag $V \in Tags$) in some protocol session $i$ and sends a message $m$ of its choice to an instance of some tag $V \in Tags$ (respectively reader $U$ $\in$*Readers*). Furthermore the adversary $A$ is allowed to block or alert the message $m$ that is sent from $U$ to $V$ (respectively $V$ to $U$) in session $i$ of a truthful protocol execution.
- Corrupt$(T, K')$ query. This query allows the adversary $A$ to learn the stored secret $K$ of the tag $T \in Tags$, and which further sets the stored secret to *K'*. Corrupt query means that the adversary has physical access to the tag, i.e., the adversary can read and tamper with the tag's permanent memory.
- Test $(i, T_0, T_1)$ query. This query does not correspond to any of $A$'s abilities, but it is necessary to define the untraceability test. When this query is invoked for session $i$, a random bit $b \in \{0, 1\}$ is generated and then, $A$ is given $T_b \in (T_0, T_1)$. Informally, $A$ wins if he can guess the bit $b$.

*Untraceable privacy (UPriv)* is defined using the game played between an adversary $A$ and a collection of the reader and the tag's instances. This game is divided into three phases:

- **Learning phase:** $A$ is able to send any Execute, Send, and Corrupt queries at will.

- **Challenge phase:** $A$ chooses two fresh tags $T_0$, $T_1$ to be tested and sends a Test query corresponding to the test session. Depending on a randomly chosen bit $b \in \{0, 1\}$, $E$ is given a tag $T_b$ from the set $\{T_0, T_1\}$. $E$ continues making any Execute, and Send queries at will.
- **Guess phase:** finally, $A$ terminates the game and outputs a bit $b' \in\{0, 1\}$, which is its guess of the value of $b$.

The success of $A$ in winning the game and thus breaking the notion of UPriv is quantified in terms of $A$'s advantage in distinguishing whether $A$ received $T_0$ or $T_1$, in other term, it correctly guessing $b$. and denoted by $Adv_A^{UPriv}(k)$ where $k$ is the security parameter.

We use the Ouafi-Phan model to verifying the achievement of untraceability property in our proposed protocol. At session $(i)$, by the Execute query, the adversary $A$ eavesdrops a perfect session between $T_0$ and a legitimate reader. He obtains the values $DID_i \oplus e_i$ and $g(N_{Ri} \parallel x_i \parallel DID_i)$. At next session, an intruder cannot replay a previously used $g(N_R \parallel x \parallel DID)$ and $DID \oplus e$ to a reader, since with high probability, it will not match the $N_R$ value generated by the reader for that session. There are two mechanisms to against the replay. Firstly, by generating an error vector with dynamic length $t' \leq t$ where $t'$ is confidential. Secondly, we accept the principle of dynamic codeword, which is stored in tag in the form of $DID$. In each session, the transmitted encoding codeword is different from the codeword of the last session because the value of the codeword is updated in the server and in the tag before the end of the session.

In addition, the security of our protocol is based on security of randomized McEliece. Nojima et al. [24] prove that padding the plaintext (in our protocol, identifier of tag $id$) with a random bit-string (random number $r$) provides the semantic security against chosen plaintext attack (IND-CPA) for the McEliece cryptosystem with the standard assumptions. So, The randomized McEliece cryptosystem is IND-CPA secure, that means if no probabilistic polynomial-time adversary wins the IND-CPA experiment with an advantage greater than a negligible function of the security parameter.

### 5.3 Informally Security Analysis

*Desynchronization resilience* In our protocol the value of the dynamic identifier $DID$ is updated in each session. This implicates a possibility of attack on desynchronization. To achieve this property, we used two secret synchronisation codewords, $c_{r_{old}}$ and $c_{r_{new}}$ stored in the server. In case the last message of the reader's authentication is blocked by the intruder, then the server updates the values of $c_{r_{old}}$ and $c_{r_{new}}$ but the tag does not update $DID$ where $DID = c_{id} \oplus c_r$. In the next session, we mention a problem in the tag's authentication with $c_{r_{new}}$, but the problem is resolved with $c_{r_{old}}$, then the tag's authentication is successful.

*Forward secrecy* Before terminating a session of protocol, the dynamic identifier $DID$ updated by using error-correcting code. The new $DID$ is $r'\mathcal{G}_1 + id\mathcal{G}_2$, where $r'$ is generated randomly in each session. The intruder could not acquire the previous dynamic identifier $DID$ used in the prior sessions. Thus, the proposed RFID authentication protocol could provide forward secrecy.

## 6 Performance Analysis

The performance of authentication protocols is mainly measured by storage space on tag, computation cost in tag and server and communications cost between the tag and the reader. Our comparison is articulated on authentication phase for each protocol. Table 3 shows the performance comparison between our protocol and the RFID protocols based on error-correcting codes.

Concerning the storage cost, the tags of protocols [11, 28, 30] require public-key matrix which is of important size compared to resources of low-cost tags. The data stored on tags of protocols [8, 26] are multiple in an agreed number of sessions. Our protocol requires only information which is dynamic identifier $DID$, thus less space is required than in other protocols.

The communication cost between a tag and a reader consists of: the number of message exchanges, and the total bit size of the transmitted messages, per each communication. Concerning our protocol, the total of the bits of the messages of communication is $2(n + l_p)$.

Concerning the computation cost, the tag requires simple operations: pseudo-random number generator and xor operation. We used the PRNG to generate $x$ and to compute $g(.)$, it is very fast. For optimising the cost of calculation of $g(.)$, we used $x$ in $g(N_R \parallel x \parallel DID)$ because the binary length of $x$ is less binary length of the error vector $e$. Concerning the server, we store the values of $c_{r_{old}}$ and $c_{r_{new}}$ instead of $r_{old}$ and $_{r_{new}}$ to augment the speed of computation in authentication phases and in the updating of $DID$. Our protocol does not need an exhaustive search for obtaining the value of $id$.

With regard to the other protocols and consideration of mutual authentication, the performance of our protocol is effective.

If we select a binary Goppa code $\mathcal{C}[n = 2048, k = 1751, d = 56]$, these parameters agree with the parameters of a secure McEliece cryptosystem for $2^{80}$ security [4]. We choose the values of $k_1$= 890 and $k_2$= 875 which are suitable with condition $k_2 < k_1$. So, the number of tags supported is $2^{875}$ tags and the space memory required in the tag is 2048 bits for codeword $DID$ and the maximal weight of the error vector is 27 bits. With these parameters, we can implement our protocol in low-cost tags, such as Mifare Classic 1K and Mifare Plus support space memory 1KB to 4 KB [21]. We note here that it is possible to optimize the parameters of the code using the techniques of Quasi-cyclic codes [3] or Quasi-dyadic codes [22]. Using the optimized parameters, we can implement our protocol in Mifare Ultralight EV1 tag support 384 bits to 1024 bits. Though several attacks can be realized against McEliece with Quasi-cyclic codes and Quasi-dyadic codes [14, 15],

Table 3: Performance Evaluation

| | Key space | Cost | | Communication | |
|---|---|---|---|---|---|
| | | Tag | Server | $T \rightarrow R$ | $R \rightarrow T$ |
| Park [26] | $l_p + n + 2\,\lvert key \rvert$ | $1P$ | $iH + 1D + 1ED$ | $n$ | - |
| Chien and Laih [9] | $n + 2\,\lvert key \rvert$ | $8P$ | $4P + 2ED$ | $2l_p + 2n)$ | $2l_p$ |
| Cui et al. [11] | $(n-k) \times (n_2 + 1)$ | $2P + 1EC$ | $2P + 1ED$ | $(n-k) + l_p$ | $l_p$ |
| Sekino et al. [28] | $(n-k) + (n-k) \times (n_1 - (n-k)/t$ | $1EC + 2P$ | $2P + 1ED$ | $(n-k) + l_p$ | $l_p$ |
| Malek and Miri [19] | $(n + k_2 + \lvert key \rvert)$ | $2P$+$CM$ | $2P + 1ED$ | $n$ | $2n + \lvert key \rvert + lp$ |
| Our Protocol | $n$ | $3P$ | $2P + 1ED$ | $n + l_p$ | $n + l_p$ |

$|key|$: length of $key$ or $id$

$i$: number of authorised sessions

$l_p$: length of generating random number or hash

$P$, $D$ and $CM$: cost of RNG or hash function, decryption operation and generation of circular matrix, respectively

$EC$ and $ED$: encoding operation and decoding operation, respectively

variants based on binary Goppa codes are secure like [6].

# 7 Conclusion

In this paper, we have discussed the limitations and vulnerabilities of previous RFID authentication protocols based on error-correcting codes. We have proposed an improved RFID authentication protocol based on randomized McEliece cryptosystem with mutual authentication, untraceability, desychronisation relisience and forward secrecy. Using formal models, the AVISPA tools and Ouafi-Phan model, we have proved security and privacy properties.

With regard to the different existing protocols based on error-correcting codes, the performance of our protocol is effective, required only $n$ bits on the tag, does not need to do exhaustive search, and the tag can perform lightweight cryptographic operations.

# References

[1] A. Armando, et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proceedings of 17th International Conference on Computer Aided Verification* (K. Etessami and S. Rajamani, eds.), vol. 3576, pp. 281–285, 2005.

[2] The AVISPA Team, "HLPSL tutorial the Beginner's guide to modelling and analysing internet security protocols," Technical Report, AVISPA project, 2006.

[3] T. P. Berger, P. L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key lengths with QC alternant codes," in *Proceedings of Africacrypt 2009*, vol. 5580, pp. 77–97, 2009.

[4] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *2nd International Workshop on Post-Quantum Cryptography (PQCRYPTO'08)*, LNCS 5299, pp. 31–46, 2008.

[5] T. Cao and P. Shen, "Cryptanalysis of two RFID authentication protocols," *International Journal of Network Security*, vol. 9, pp. 95–100, 2009.

[6] P. L. Cayrel, G. Hoffmann, and E. Persichetti, "Efficient implementation of a CCA2-secure variant of McEliece using generalized Srivastava codes," in *Proceedings of PKC'2012*, LNCS 7293, pp. 138–155, 2012.

[7] C. L. Chen, Y. L. Lai, C. C. Chen, Y. Y. Deng, and Y. C. Hwang, "RFID ownership transfer authorization systems conforming EPCglobal class-1 generation-2 standards," *International Journal of Network Security*, vol. 13, pp. 41–48, 2011.

[8] H. Y. Chien, "Secure access control schemes for RFID systems with anonymity," in *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, pp. 96, 2006.

[9] H. Y. Chien and C. S. Laih, "ECC-based lightweight authentication protocol with untraceability for low-cost RFID," *Journal of Parallel and Distributed Computing*, vol. 69, pp. 848–853, 2009.

[10] T. M. Cover, "Enumerative source encoding," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, 1973.

[11] Y. Cui, K. Kobara, K. Matsuura, and H. Imai, "Lightweight asymmetric privacy-preserving authentication protocols secure against active attack," in *Proceedings of the Fifth Annual IEEE International Conference (PerComW'07)*, pp. 223–228, 2007.

[12] V. Deursen, S. Mauw, and S. Radomirovic, "Untraceability of RFID protocols," *Information Security Theory and Practices, Smart Devices, Convergence and Next Generation Networks*, pp. 1–15, 2008.

[13] D. Dolev and A. C. Yao, "On security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, pp. 198–208, 1983.

[14] J. C. Faugère, A. Otmani, L. Perret, and J. P. Tillich, "Algebraic cryptanalysis of McEliece variants with compact keys," in *Proceedings of the 29th International Conference on Cryptology (EUROCRYPT'10 2010)*, pp. 279-298, 2010.

[15] J. C. Faugère, A. Otmani, L. Perret, and J. P. Tillich, "A distinguisher for high rate McEliece cryptosystems," Technical Report 2010/331, Cryptology ePrint Archive, 2010.

[16] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Transactions on Information and System Security*, vol. 13, no. 1, Oct. 2009.

[17] W. Khedr, "On the Security of Moessner's and Khan's Authentication Scheme for Passive EPC-global C1G2 RFID Tags," *International Journal of Network Security*, vol. 16, no. 5, pp. 369–375, 2014.

[18] K. Kobara and H. Imai, "Personalized-public-key cryptosystem (P2KC) - Application where public-key size of Niederreiter PKC can be reduced," in *Workshop on Codes and Lattices in Cryptography (CLC'06)*, pp. 61–68, 2006.

[19] B. Malek and A. Miri, "Lightweight mutual RFID authentication," in *Proceedings of IEEE International Conference on Communications*, pp. 868–872, 2012.

[20] R. J. McEliece, "A public-key system based on algebraic coding theory," Technical Report, DSN Progress Report 44, Jet Propulsion Lab, 1978.

[21] *The Mifar Cards*, 2015. (http://www.mifare.net)

[22] R. Misoczki and P. S. L. M. Barreto, "Compact McEliece keys from Goppa codes," in *Selected Areas in Cryptography (SAC'09)*, LNCS, pp. 376–392, 2009.

[23] M. Naveed, W. Habib, U. Masud, U. Ullah, and G. Ahmad, "Reliable and low cost RFID based authentication system for large scale deployment," *International Journal of Network Security*, vol. 14, pp. 173–170, 2012.

[24] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Designs, Codes and Cryptography*, vol. 49, no. 1–3, pp. 289–305, 2008.

[25] K. Ouafi and R. C. W. Phan, "Privacy of recent RFID authentication protocols," in *Proceedings of ISPEC'08* (L. Chen, Y. Mu, and W. Susilo, eds.), LNCS 4991, pp. 263–277, 2008.

[26] C. S. Park, "Authentication protocol providing user anonymity and untraceability in wireless mobile communication systems," *Computer Networks*, vol. 44, pp. 267–273, 2004.

[27] P. M. Schalkwijk, "An algorithm for source coding," *IEEE Transactions of Information Theory*, vol. 18, no. 3, pp. 395–399, 1972.

[28] T. Sekino, Y. Cui, K. Kobara, and H. Imai, "Privacy enhanced RFID using Quasi-Dyadic fix domain shrinking," in *Proceedings of Global Telecommunications Conference (GLOBECOM'10)*, pp. 1–5, 2010.

[29] N. Sendrier, "Encoding information into constant weight words," in *IEEE Conference on ISIT'05*, pp. 435–438, 2005.

[30] M. Suzuki, K. Kobara, and H. Imai, "Privacy enhanced and light weight RFID system without tag synchronization and exhaustive search," in *Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1250–1255, 2006.

[31] P. Véron, "Code based cryptography and steganography," in *Proceedings of CAI'13*, LNCS 8080, pp. 9–46, 2013.

[32] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "An authentication protocol for low-cost RFID tags," *International Journal of Mobile Communications*, vol. 9, no. 2, pp. 208–223, 2011.

[33] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "A mutual authentication protocol for RFID," *IEEE IT Professional*, vol. 13, no. 2, pp. 20–24, 2011.

[34] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "An improved authentication protocol for mobile agent device in RFID," *International Journal of Mobile Communications*, vol. 10, no. 5, pp. 508–520, 2012.

[35] X. Zhang and B. King, "Security requirements for RFID computing systems," *International Journal of Network Security*, vol. 6, pp. 214–226, 2008.

**Noureddine Chikouche**, received his B.Sc. (engineer) in computer science from the University of Constantine, Algeria, in 1999. In addition, he received his M.Sc. in computer science from the University of M'sila, Algeria, in 2010. He has been a Ph.D. candidate at University of Biskra, Algeria. He also is assistant professor in computer sciences Department at University of M'sila, from 2011. His research interests include RFID security, formal verification of cryptographic protocols, and code-based cryptography.

**Foudil Cherif** is an associate professor of computer science at Computer Science Department, Biskra University, Algeria. Dr. Cherif holds Ph.D degree in computer science. The topic of his dissertation is behavioral animation: crowd simulation of virtual humans. He also possesses B.Sc. (engineer) in computer science from Constantine University 1985, and an M.Sc. in computer science from Bristol University, UK in 1989. He is currently the head of LESIA Laboratory. His current research interest is in Artificial intelligence, Artificial life, Crowd simulation, RFID security, formal verification of cryptographic protocols and Software engineering.

**Pierre-Louis Cayrel**, received his Ph.D. degree in Mathematics from University of Limoges in 2008. He has been a post-doctorate assistant in CASED in Darmstadt, Germany from 2009 to 2011. He is now an Associate Professor in Jean Monnet University, Saint-Etienne since September 2011. His research interests are: coding theory, code-based cryptography, side channel analysis and secure implementations of cryptographic schemes.

**Mohamed Benmohammed**, received his Ph.D. degree in computer science from University of Sidi Bel Abbès, Algeria in 1997. He is currently a professor in the computer science Department, University of Constantine 2, Algeria. His research interests are Micropocessor Architecture, Embedded systems, and real time applications.

# Global Analysis of a SEIQV Epidemic Model for Scanning Worms with Quarantine Strategy

Fangwei Wang[1,2], Yong Yang[3], Yunkai Zhang[4] and Jianfeng Ma[2]
*(Corresponding author: Yunkai Zhang)*

College of Information Technology, Hebei Normal University[1]
No. 20, South ErHuan Road, YuHua District, Shijiazhuang 050024, China
School of Telecommunications Engineering, Xidian University[2]
No. 2, South Taibai Road, YanTa District, Xi'an, Shaanxi 710071, China
Network Information Center, Yunnan University[3]
No. 2, North CuiHu Road, WuHua District, Kunming 650091, China
NDepartment of Information Engineering, Shijiazhuang Institute of Railway Technology[4]
No. 18, Sishuichang Road, Qiaodong District, Shijiazhuang 050071, China
(Email: zhyk@hebtu.edu.cn)

*(Received Oct. 10, 2013; revised and accepted Mar. 13 & May 21, 2014)*

## Abstract

Active scanning worms have drawn a significant attention due to their enormous threats to the Internet infrastructure and services. In order to effectively defend against them, this paper proposes a novel epidemic SEIQV model with quarantine strategy. Using this SEIQV model, we obtain the basic reproduction number for determining whether the worm dies out completely. The global stabilities of worm-free equilibrium and endemic equilibrium are proved, and determined by the basic reproduction number. The impact of different parameters of this model is studied. Simulation results show that the number of susceptible, infected and vaccinated hosts are consistent with theoretical analysis. The model provides a theoretical foundation for controlling and forecasting for active scanning worms.

*Keywords: Basic reproduction number, network security, quarantine strategy, stability analysis, worm propagation model*

## 1 Introduction

Active scanning worms are malicious codes which can replicate themselves and actively infected other hosts with certain vulnerability via Internet. With the ever increasing number of Internet applications and the emergence of new technologies, worms have become a great threat to our work and daily life, caused tremendous economic losses. Especially, the advent of the Internet of things would make the threat increasingly serious. How to combat Internet worms effectively is an urgent issue confronted with defenders. Therefore, it is necessary to comprehend the long-term behavior of worms and to propose effective strategies to defend against worms.

Based on the infectivity between a worm and a biological virus, some epidemic models representing worm propagations were presented to depict the propagation of worms, e.g., *SIR* model [15], *SIRS* model [9, 14], *SIQV* model [24], *SIDQV* model [25], which all assume that susceptible hosts can immediately translate into infected ones. This assumption is unreasonable. Actually, it will take a certain time to send worm copies to susceptible hosts. To overcome previous drawbacks, some researchers added a state ($E$), namely the exposed state, and then proposed some propagation models, e.g., *SEIR* model [10], *SEIRS* model [11, 13, 17], *SEIQV* model [18], *SEIQRS* model [8], which assume that exposed hosts can not infect other ones. Actually, an infected host which is in latency can infect other hosts by means of some methods, e.g., vulnerability seeking. All the previous models do not take this passive infectivity into consideration. Recently, Yang et al. [20, 21, 22, 23] proposed some models, by taking into account the fact that a host immediately possesses infectivity once it is infected. These model, however, all make an assumption that exposed hosts and infected hosts have the same infectivity. This is not consistent with the reality. Although an exposed host also sends scanning packets to find susceptive hosts with certain vulnerabilities, the scanning packets sent by an exposed host are less than an infected one. Usually, the infection rate of exposed hosts is less than that of infected ones. Therefore, they should have different infection rates.

Recently, more attention has been paid to the combination of worm propagation model and countermeasures to study the prevalence of worms, e.g., quarantine [24, 25] and vaccination [4]. The implementation of quarantine strategy relies on the intrusion detection systems (IDSes). Intrusion detection systems can be classified into two categories: misuse and anomaly intrusion detection systems. The former is mainly based on a database with the feature of known attack behaviors, which fails to detect new ones. The latter can detect both novel and known worms, but false positive rate is high. In summary, both classes of intrusion detection systems have defects that affect their performances.

*SEIQV* model [18] takes quarantine and vaccination into consideration as the two main recovery countermeasures, and analyzes the global stability of its worm-free equilibrium. Inspired by *SEIQV* model [18], we propose a new extended model, referred to as e-SEIQV (susceptible - exposed - infected - quarantined - vaccinated) model. In comparison with *SEIQV* model [18], the model proposed takes two infection rates into account. Some susceptible hosts can be directly vaccinated. Using the basic reproduction number, we derive global stabilities of a worm-free equilibrium and a unique endemic equilibrium by a Lyapunov function and a geometric approach. Based on these results and further analysis, some effective methods for controlling worms are recommended.

The rest of this paper is organized as follows. Section 2 formulates the new model and obtains its basic reproduction number. Section 3 proves the local and global stabilities of the worm-free equilibrium. Section 4 examines the local and global stabilities of the endemic equilibrium. Section 5 covers the numerical analysis and the simulations. Section 6 summarizes the paper with some future directions.

## 2 Mathematical Model Formulation

The total host population $N$ is partitioned into five groups and any hosts can potential be in any of these groups at any time $t$: the susceptible, exposed, infected, quarantined, vaccinated, with sizes denoted by $S$, $E$, $I$, $Q$, $V$, respectively. The total number of population $N$ at time $t$ is given by $N(t) = S(t) + E(t) + I(t) + Q(t) + V(t)$. The dynamical transfer of hosts is depicted in Figure 1.

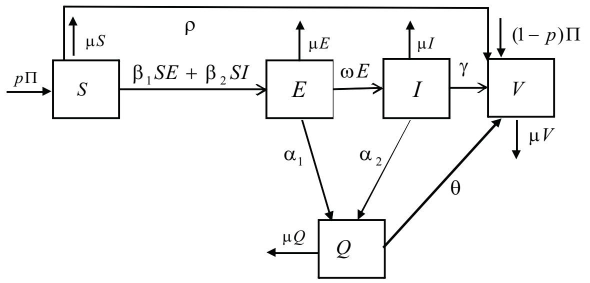


Figure 1: State transition diagram with quarantine

In the model, susceptible hosts can be infected by worms with efficient infection rates $\beta_1$, $\beta_2$ and become into exposed ones and infected ones, or patched into the vaccinated state with rate $\rho$. $\omega$ is the transfer rate between the exposed and the infected. Some exposed and infected ones can be detected by a misuse detection system and then constantly quarantined at rates $\alpha_1$, $\alpha_2$, respectively. $\alpha_1$, $\alpha_2$ are determined by the misuse detection system, which will become larger if the detection system is set to be sensitive to worms' activities. A high performance detection system has higher detection rate and lower false alarm rate. For example, the detection system should have larger $\alpha_1$, $\alpha_2$. Some hosts in the quarantined state become vaccinated ones by repairing and then patching at rate $\theta$. Some infected hosts can be detected and then manually patched at rate $\gamma$. The positive parameter $\mu$ is the death rate in each of the five states. Some hosts enter the network at the number $\Pi$, a fraction $1 - p$ of which is patched into the vaccinated state directly at "birth".

Based on the compartment model presented Figure 1, our model having infected force in the exposed, infected period is described by the following system of differential equations:

$$\begin{cases} S^{'}(t) = p\Pi - \beta_1 SE - \beta_2 SI - (\rho + \mu)S, \\ E^{'}(t) = \beta_1 SE + \beta_2 SI - (\omega + \alpha_1 + \mu)E, \\ I^{'}(t) = \omega E - (\gamma + \alpha_2 + \mu)I, \\ Q^{'}(t) = \alpha_1 E + \alpha_2 I - (\theta + \mu)Q, \\ V^{'}(t) = \rho S + \gamma I + \theta Q + (1-p)\Pi - \mu V. \end{cases} \tag{1}$$

Summing the equations of System (1), we obtain

$$N(t)^{'} = \Pi - \mu(S + E + I + Q + V). \tag{2}$$

Therefore, the total population size $N(t)$ converges to the equilibrium $\Pi/\mu$. It follows from Equation (2) that $\liminf_{t\to\infty} N(t) \leq \Pi/\mu$. We thus study our System (1), in the following feasible region:

$$\Omega = \{(S, E, I, Q, V) \in \mathbb{R}^5_+ : S + E + I + Q + V \leq \Pi/\mu\},$$

which is a positively invariant set of Model (1). We next consider the dynamic behavior of Model (1) on $\Omega$. It is easy to see that Model (1) always has a worm-free equilibrium, $P_0 = (p\Pi/(\rho + \mu), 0, 0, 0, \frac{\Pi}{\mu}(1 - \frac{p}{\rho+\mu}))$.

Let $x = (E, I, Q, V, S)^T$, then Model (1) can be written as

$$\frac{dx}{dt} = \mathcal{F}(x) - \mathcal{V}(x),$$

where

$$\mathcal{F}(x) = \begin{pmatrix} \beta_1 SE + \beta_2 SI \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\mathcal{V}(x) = \begin{pmatrix} (\omega+\alpha_1+\mu)E \\ (\gamma+\alpha_2+\mu)I-\omega E \\ (\theta+\mu)Q-\alpha_1 E-\alpha_2 I \\ \mu V-\rho S-\gamma I-\theta Q-(1-p)\Pi \\ \beta_1 SE+\beta_2 SI+(\rho+\mu)S-p\Pi \end{pmatrix}.$$

Differentiating $\mathcal{F}(x)$ and $\mathcal{V}(x)$ with respect to $E, I, Q, V, S$ and evaluating at the worm-free equilibrium $P_0 = (p\Pi/(\rho+\mu), 0, 0, 0, \frac{\Pi}{\mu}(1-\frac{p}{\rho+\mu}))$, respectively, we have

$$D\mathcal{F}(P_0) = \begin{pmatrix} F_{2\times2} & 0_{2\times3} \\ 0_{3\times2} & 0_{3\times3} \end{pmatrix},$$

$$D\mathcal{V}(P_0) = \begin{pmatrix} Y_{2\times2} & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ Y'_{3\times2} & \theta+\mu & 0 & 0 \\ & -\theta & \mu & -\rho \\ & 0 & 0 & \rho+\mu \end{pmatrix},$$

where

$$F_{2\times2} = \begin{pmatrix} \beta_1 p\Pi/(\rho+\mu) & \beta_2 p\Pi/(\rho+\mu) \\ 0 & 0 \end{pmatrix},$$

$$Y'_{3\times2} = \begin{pmatrix} -\alpha_1 & -\alpha_2 \\ 0 & -\gamma \\ \beta_1 p\Pi/(\rho+\mu) & \beta_2 p\Pi/(\rho+\mu) \end{pmatrix},$$

and

$$Y_{2\times2} = \begin{pmatrix} \omega+\alpha_1+\mu & 0 \\ -\omega & \gamma+\alpha_2+\mu \end{pmatrix}.$$

$\mathcal{F}\mathcal{V}^{-1}$ is the next generation matrix for Model (1). Thus, the spectral radius of the next generation matrix $\mathcal{F}\mathcal{V}^{-1}$ can be obtained as,

$$\begin{aligned} \rho(\mathcal{F}\mathcal{V}^{-1}) &= \rho(F_{2\times2}Y_{2\times2}^{-1}) \\ &= \tfrac{p\Pi(\beta_1(\gamma+\alpha_2+\mu)+\beta_2\omega)}{(\rho+\mu)(\omega+\alpha_1+\mu)(\gamma+\alpha_2+\mu)}. \end{aligned}$$

According to Theorem 2 in [2], the basic reproduction number of Model (1) is

$$R_0 = \frac{p\Pi(\beta_1(\gamma+\alpha_2+\mu)+\beta_2\omega)}{(\rho+\mu)(\omega+\alpha_1+\mu)(\gamma+\alpha_2+\mu)}. \tag{3}$$

For the concision of notation, let $m = \omega+\alpha_1+\mu$ and $n = \gamma+\alpha_2+\mu$. Thus $R_0 = \frac{p\Pi(\beta_1 n+\beta_2\omega)}{(\rho+\mu)mn}$.

The first four equations in System (1) have no dependence on the fifth one. Therefore, the fifth equation can be omitted. Thus, System (1) can be rewritten as the following four-dimensional system:

$$\begin{cases} S'(t) = p\Pi-\beta_1 SE-\beta_2 SI-(\rho+\mu)S, \\ E'(t) = \beta_1 SE+\beta_2 SI-(\omega+\alpha_1+\mu)E, \\ I'(t) = \omega E-(\gamma+\alpha_2+\mu)I, \\ Q'(t) = \alpha_1 E+\alpha_2 I-(\theta+\mu)Q. \end{cases} \tag{4}$$

Next, we will study the stabilities of worm-free equilibrium and endemic equilibrium on System (4).

# 3 Stability of Worm-free Equilibrium

It is easily obtained that System (4) has a worm-free equilibrium given by $P_0 = (p\Pi/(\rho+\mu), 0, 0, 0)$.

**Lemma 1.** *When $R_0 < 1$, the worm-free equilibrium $P_0$ is locally asymptotically stable in $\Omega$. When $R_0 > 1$, the worm-free equilibrium $P_0$ is an unstable saddle point.*

*Proof.* The Jacobian matrix of Model (4) at $P_0$ is

$$J(P_0) = \begin{pmatrix} -(\rho+\mu) & -\frac{\beta_1 p\Pi}{(\rho+\mu)} & -\frac{\beta_2 p\Pi}{(\rho+\mu)} & 0 \\ 0 & \frac{\beta_1 p\Pi}{(\rho+\mu)}-m & \frac{\beta_2 p\Pi}{(\rho+\mu)} & 0 \\ 0 & \omega & -n & 0 \\ 0 & \alpha_1 & \alpha_2 & -(\theta+\mu) \end{pmatrix}$$

It is easily obtained that $J(P_0)$ has two negative eigenvalues $\lambda_1 = -(\rho+\mu)$, and $\lambda_2 = -(\theta+\mu)$, the other eigenvalues of $J(P_0)$ are determined by the following equation

$$\lambda^2+(m+n-\frac{\beta_1 p\Pi}{(\rho+\mu)})\lambda+mn-\frac{(n\beta_1+\omega\beta_2)p\Pi}{(\rho+\mu)} = 0. \tag{5}$$

When $R_0 < 1$, then $mn > (n\beta_1+\omega\beta_2)p\Pi/(\rho+\mu)$.

For $mn > (n\beta_1+\omega\beta_2)p\Pi/(\rho+\mu)$, we can obtain $m+n > n+\beta_1 p\Pi/(\rho+\mu)+p\Pi\beta_2\omega/(\rho+\mu)$, thus $m+n-\beta_1 p\Pi/(\rho+\mu) > n+\beta_2 p\Pi/(\rho+\mu) > 0$, which means the Equation (5) has two negative roots. Therefore, the worm-free equilibrium $P_0$ is locally asymptotically stable.

When $R_0 > 1$, then $mn-(n\beta_1+\omega\beta_2)p\Pi/(\rho+\mu) < 0$, which means the Equation (5) has a positive root and a negative root. Therefore, the worm-free equilibrium $P_0$ is unstable saddle point. □

**Lemma 2.** *When $R_0 \le 1$, the worm-free equilibrium $P_0$ is globally asymptotically stable in $\Omega$. When $R_0 > 1$, all solutions starting in $\Omega$ and sufficiently close to $P_0$ move away from $\{P_0\}$.*

*Proof.* Consider the Lyapunov function

$$L = \frac{\beta_1 n+\beta_2\omega}{mn}E+\frac{\beta_2}{n}I.$$

Its derivative along the solutions to Model (4) is

$$\begin{aligned} L' &= \tfrac{\beta_1 n+\beta_2\omega}{mn}(\beta_1 SE+\beta_2 SI-mE)+\tfrac{\beta_2}{n}(\omega E-nI) \\ &= \tfrac{\beta_1 n+\beta_2\omega}{mn}(\beta_1 SE+\beta_2 SI)-(\beta_1 E+\beta_2 I) \\ &= (\beta_1 E+\beta_2 I)(\tfrac{\beta_1 n+\beta_2\omega}{mn}S-1) \\ &\le (\beta_1 E+\beta_2 I)(\tfrac{p\Pi(\beta_1 n+\beta_2\omega)}{mn(\rho+\mu)}-1) \\ &= (\beta_1 E+\beta_2 I)(R_0-1) \\ &\le 0. \end{aligned}$$

Furthermore, $L' = 0$ if and only if $E = I = 0$ or $R_0 = 1$. Thus, the largest compact invariant set in $\{(S, E, I, Q)|L' = 0\}$ is the singleton $\{P_0\}$. When $R_0 \le 1$, the global stability of $P_0$ follows from LaSalle's invariance principle [5]. LaSalle's invariance principle [5] implies that $P_0$ is globally asymptotically stable in $\Omega$. When $R_0 > 1$, it follows from the fact $L' > 0$ if $E > 0$ and $I > 0$. This completes the proof. □

## 4 Stability of Endemic Equilibrium

The endemic equilibrium $P^*(S^*, E^*, I^*, Q^*)$ of Model (4) is determined by equations

$$\begin{cases} p\Pi - \beta_1 SE - \beta_2 SI - (\rho+\mu)S = 0, \\ \beta_1 SE + \beta_2 SI - (\omega + \alpha_1 + \mu)E = 0, \\ \omega E - (\gamma + \alpha_2 + \mu)I = 0, \\ \alpha_1 E + \alpha_2 I - (\theta + \mu)Q = 0. \end{cases} \tag{6}$$

By some simple computation, we obtain

$$\begin{cases} S^* = \frac{p\Pi}{R_0(\rho+\mu)}, \\ E^* = \frac{(\gamma+\alpha_2+\mu)I}{\omega} = \frac{nI}{\omega}, \\ I^* = \frac{p\Pi(1-1/R_0)\omega}{(\omega+\alpha_1+\mu)(\gamma+\alpha_2+\mu)} = \frac{p\Pi(1-1/R_0)\omega}{mn}, \\ Q^* = \frac{(\alpha_1(\gamma+\alpha_2+\mu)+\alpha_2\omega)I}{\omega(\theta+\mu)} = \frac{(\alpha_1 n+\alpha_2\omega)I}{\omega(\theta+\mu)}. \end{cases} \tag{7}$$

Now we investigate the local stability of the endemic equilibrium $P^*(S^*, E^*, I^*, Q^*)$. The Jacobian matrix of Equation (4) at the endemic equilibrium $P^*$ is

$$J(P^*) = \begin{pmatrix} B_1 & -\beta_1 S & -\beta_2 S & 0 \\ \beta_1 E + \beta_2 I & \beta_1 S - m & \beta_2 S & 0 \\ 0 & \omega & -n & 0 \\ 0 & \alpha_1 & \alpha_2 & -\theta-\mu \end{pmatrix} \tag{8}$$

where, $B_1 = -\beta_1 E - \beta_2 I - \rho - \mu$. Its characteristic equation is $det(\lambda I - J(P^*)) = 0$, where $I$ is the unit matrix. Therefore,

$$det(\lambda I - J(P^*)) = (\lambda_1 + \theta + \mu)(\lambda^3 + A\lambda^2 + B\lambda + C) = 0, \tag{9}$$

where

$$A = n + \rho + \mu + \beta_1 E + \beta_2 I + \frac{\beta_2 \omega m}{\beta_1 n + \beta_2 \omega} > 0,$$

$$\begin{aligned} B &= \tfrac{\beta_2 m\omega}{\beta_1 n + \beta_2 \omega}(\rho + \mu + \beta_1 E + \beta_2 I) \\ &+ \beta_1 S(\beta_1 E + \beta_2 I) + n(\rho + \mu + \beta_1 E + \beta_2 I) \\ &> 0, \end{aligned}$$

$$C = mn(\beta_1 E + \beta_2 I) > 0.$$

By a direct calculation, we obtain that $AB - C > 0$. According to the theorem of Routh-Hurwitz [1], it follows that all the roots of Equation (9) have negative real parts. Therefore, the endemic equilibrium $P^*$ is locally asymptotically stable.

From the above discussion, we can summarize the following conclusion.

**Lemma 3.** *When $R_0 > 1$, the endemic equilibrium $P^*$ is locally asymptotically stable in $\Omega$.*

Next, we apply the geometrical approach [7] to investigate the global stability of the endemic equilibrium $P^*$ in the region $\Omega$.

**Theorem 1.** [7] *Consider the following systems:*
$x^{'} = f(x)$, $x \in \Omega$.
*If the following conditions are satisfied:*

1) *The system $(*)$ exists a compact absorbing set $K \subset \Omega$ and has a unique equilibrium $P^*$ in $\Omega$;*

2) *$P^*$ is locally asymptotically stable;*

3) *The system $(*)$ satisfies a Poincaré-Bendixson criterion;*

4) *A periodic orbit of the system $(*)$ is asymptotically orbitally stable, then the only equilibrium $P^*$ is the globally asymptotically stable in $\Omega$.*

**Lemma 4.** *If $R_0 > 1$, the unique positive equilibrium $P^*$ of Model (4) is globally asymptotically stable in $\Omega$.*

*Proof.* We only need to prove that all assumptions of Theorem 1 hold.

If $R_0 > 1$, then the worm-free equilibrium is unstable according to Lemma 1. Moreover, the behavior of the local dynamics near the region $P_0$ described in Lemma 1 implies that Model (4) is uniformly persistent in the region $\Omega$. That is, there exists a constant $c > 0$, such that any solution $(S(t), E(t), I(t), Q(t))$ of Model (4) with initial value $(S(0), E(0), I(0), Q(0))$ in $\Omega$ satisfies

$$min\{\liminf_{t\to\infty} S(t), \liminf_{t\to\infty} E(t), \liminf_{t\to\infty} I(t), \liminf_{t\to\infty} Q(t)\} \geq c.$$

This can be proved by applying a uniform persistent result in [3] and by the use of a similar argument as in the proof in [6]. The uniform persistence of System (4) in the bounded set $\Omega$ is equivalent to the existence of a compact $K \in \Omega$ that is absorbing for System (4). During the process of obtaining the endemic equilibrium $P^*$, we can know that $P^*$ is the unique equilibrium in the interval $(0, \Pi/\mu)$. Assumption (1) holds.

According to Lemma 3, we know that the endemic equilibrium $P^*$ is locally asymptotically stable in the region $\Omega$. Assumption (2) holds.

The Jacobian matrix of Model (4) is denoted by Equation (8). Choosing the matrix $H$ as $H = diag(-1, 1, -1, -1)$, it is easy to prove that $HJH$ has non-positive off-diagonal elements, thus we can obtain that System (4) is competitive. This verifies the Assumption (3).

The second compound matrix $J^{[2]}(P^*)$ of $J(P^*)$ can be calculated as follows:

$$J^{[2]}(P^*) = \begin{pmatrix} A1 & \beta_2 S & 0 & \beta_2 S & 0 & 0 \\ \omega & A2 & 0 & -\beta_1 S & 0 & 0 \\ \alpha_1 & \alpha_2 & A3 & 0 & -\beta_1 S & -\beta_2 S \\ 0 & b & 0 & A4 & 0 & 0 \\ 0 & 0 & b & \alpha_2 & A5 & \beta_2 S \\ 0 & 0 & 0 & -\alpha_1 & \omega & A6 \end{pmatrix} \tag{10}$$

where,
$A1 = -(\beta_1 E + \beta_2 I + \rho + \mu + m - \beta_1 S)$,

$$\begin{aligned}
&A2 = -(\beta_1 E + \beta_2 I + \rho + \mu + n),\\
&A3 = -(\beta_1 E + \beta_2 I + \rho + 2\mu + \theta),\\
&A4 = -(m + n - \beta_1 S),\\
&A5 = -(m + \mu + \theta - \beta_1 S),\\
&A6 = -(n + \theta + \mu),\\
&b = \beta_1 E + \beta_2 I.
\end{aligned}$$

The second compound system of Model (4) in a periodic solution can be represented by the following differential equations:

$$\begin{cases}
X^{'}(t) = A1X + \beta_2 SY + \beta_2 SL,\\
Y^{'}(t) = \omega X + A2Y - \beta_1 SL,\\
Z^{'}(t) = \alpha_1 X + \alpha_2 Y + A3Z - \beta_1 SM - \beta_2 SU,\\
L^{'}(t) = bY + (\beta_1 S - m - n)L,\\
M^{'}(t) = bZ + \alpha_2 L + A5M + \beta_2 SU,\\
U^{'}(t) = -\alpha_1 L + \omega M - (n + \mu + \theta)U.
\end{cases} \tag{11}$$

In order to prove that System (11) is asymptotically stable, we consider the following Lyapunov function:

$$\begin{aligned}
& V(X, Y, Z, L, M, U; S, E, I, Q)\\
= \;& \sup\{|X| + |L| + |M|, \frac{E}{I}(|Y| + |Z| + |U|)\}.
\end{aligned}$$

By the use of the uniform persistence, we obtain that the orbit of $P(t) = (S(t), E(t), I(t), Q(t))$ remains a positive distance from the boundary of $\Omega$, thus, we know that there exists a constant $c$ satisfying

$$\begin{aligned}
& V(X, Y, Z, L, M, U; S, E, I, Q)\\
\geq \;& c \sup\{|X|, |Y|, |Z|, |L|, |M|, |U|\},
\end{aligned}$$

for all $(X, Y, Z, L, M, U) \in \mathbb{R}^6$ and $(S, E, I, Q) \in P(t)$.

For the differential equations in Equation (11), we can obtain the following differential inequalities by direct calculations:

$$\begin{aligned}
[D_+(|X| + |Y| + |Z|) \;\leq\;& -(2\mu + \omega + \alpha_1)(|X| + |L|\\
& +|M|) + \frac{E}{I}(\beta_1 S + \beta_2 S\frac{I}{E})\\
& (|Y| + |Z| + |U|),\\
D_+(|L| + |M| + |U|) \;\leq\;& \omega(|X| + |L| + |M|) - (2\mu\\
& +\alpha_2 + \gamma)(|Y| + |Z| + |U|).
\end{aligned}$$

Then,

$$\begin{aligned}
D_+\frac{E}{I}(|Y| + |Z| + |U|) \;\leq\;& \omega\frac{E}{I}(|X| + |L| + |M|)\\
& +(\frac{E^{'}}{E} - \frac{I^{'}}{I} - (2\mu + \alpha_2\\
& +\gamma))\frac{E}{I}(|Y| + |Z| + |U|).
\end{aligned}$$

From the pervious formula, we can obtain

$$D_+|V(t)| \leq \max\{g_1(t), g_2(t)\}V(t),$$

where,

$$\begin{aligned}
g_1(t) &= -(2\mu + \delta_1 + \omega) + (\beta_1 S + \beta_2 S\frac{I}{E}),\\
g_2(t) &= \omega\frac{E}{I} + \frac{E^{'}}{E} - \frac{I^{'}}{I} - (2\mu + \alpha + \delta_2 + p).
\end{aligned}$$

From Model (4), we can obtain

$$\begin{aligned}
\frac{E^{'}}{E} &= \beta_1 S + \beta_2 S\frac{I}{E} - (\omega + \alpha_1 + \mu),\\
\frac{I^{'}}{I} &= \omega\frac{E}{I} - (\gamma + \alpha_2 + \mu).
\end{aligned}$$

Therefore,

$$g_1(t) = \frac{E^{'}}{E} - \mu, g_2(t) = \frac{E^{'}}{E} - \mu.$$

Then,

$$\int_0^{\zeta} \sup\{g_1(t), g_2(t)\}dt \leq \ln E(t)|_0^{\zeta} - \mu\zeta = -\mu\zeta < 0,$$

which implies that $(X(t), Y(t), Z(t), L(t), M(t), U(t)) \to 0$, as $t \to \infty$. Thus, the second compound System (11) is asymptotically stable. This verifies the Assumption (4).

We verify all the assumptions of Theorem 1. Therefore, $P^*$ is globally asymptotically stable in $\Omega$. □

## 5 Numerical Simulations

In this experiment, we choose the Slammer as basic behavior of a worm. To obtain the spread of worms in a large-scale network, 1,000,000 hosts are selected as the population size. According to the real conditions of the Slammer worm, the worm's average scan rate is $s = 4000$ per second [12]. Slammer worm's infection rate can then be computed as $\beta_2 = s/2^{32} = 0.00000093$, $\beta_1 = 0.0000009$. At the beginning, the number of susceptible, exposed, infected, quarantined and vaccinated hosts are $S(0) = 999,990$, $E(0) = 0$, $I(0) = 10$, $Q(0) = 0$ and $V(0) = 0$, respectively. The quarantined rates of exposed hosts and infected hosts are $\alpha_1 = 0.0001$ , $\alpha_2 = 0.004$ per minute, respectively.

Other parameters in these simulations are given as follows: $\mu = 0.00001$, $\rho = 0.00002$, $\theta = 0.005$, $\omega = 0.05$, $p = 0.1$, $\gamma = 0.001$, where $R_0 = 0.677 < 1$. The worm will gradually disappear according to Theory 2. Figure 2 illustrates the number of susceptible, exposed and infected hosts when $R_0$ is 0.677. From Figure 2, we can clearly see that the tendency of the worm propagation is depressive, which is consistent with Lemma 2. Finally, all infected hosts vanish, and become into vaccinated state. In order to effectively defend against such worms, we must adopt some feasible methods to decrease the infection rate [16, 19] or increase the following parameters (e.g., the transfer rates between the exposed and the recovered, between the exposed and the infected) to guarantee the basic reproduction number $R_0 < 1$.

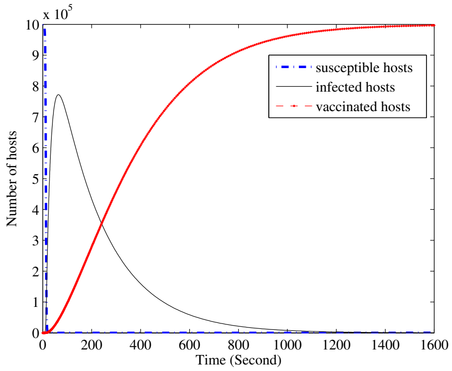


Figure 2: Globally asymptotically stable worm-free equilibrium

When $\alpha_1 = 0.02$, $\alpha_2 = 0.06$, $\theta = 0.009$, we can obtain $R_0 = 2.654 > 1$. Other parameters do not vary. We can see the results in Figure 3. As can be seen from Figure 3, the number of susceptible and infected hosts eventually become positive values between 0 and $\Pi/\mu$. $S(t)$, $I(t)$, $V(t)$ all approach their steady state, and the worm persists. This is fully consistent with the conclusions of Lemma 4.

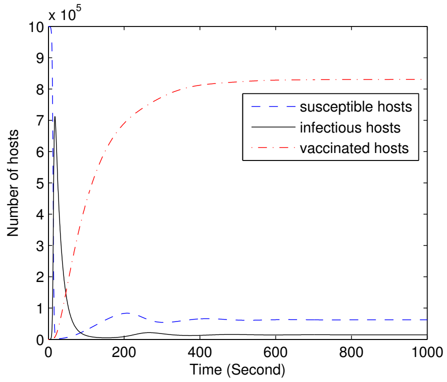


Figure 3: Globally asymptotically stable endemic equilibrium

In our model, the main defending method is the use of quarantine strategy. There are two quarantined rates in the proposed model. Intuitively, both of them all play an important role in decreasing the number of infected hosts. Next, we study the effect of the quarantined rates. when $\alpha_1$ is equal to 0.0001, 0.001 and 0.01, respectively, we can see the result in Figure 4. From Figure 4, we can see that the quarantined rate $\alpha_1$ plays a minor role in decreasing the number of hosts infected by worms. On the other hand, a larger $\alpha_1$ might cause a higher false alarm rate of the detection system, and block some users' normal activities.

When we change the values of the quarantined rate $\alpha_2$, e.g., 0.002, 0.004, 0.006 and 0.008, we obtain the result in Figure 5. From Figure 5, it can be seen that the quarantined rate $\alpha_2$ has an obvious significant effect on defending worms. The larger the quarantined rate is, the less the number of infected hosts are. The quarantined rate plays an important role in containing the infected hosts. The quarantined rate relies mainly on the accuracy of intrusion detection systems. The detection rate depends mainly on the quarantined rate $\alpha_2$ of infected hosts. That is, a more effective detection rate will generate a larger quarantined rate. We can improve the efficiency and decrease the false positive of intrusion detection systems to obtain a larger quarantined rate.

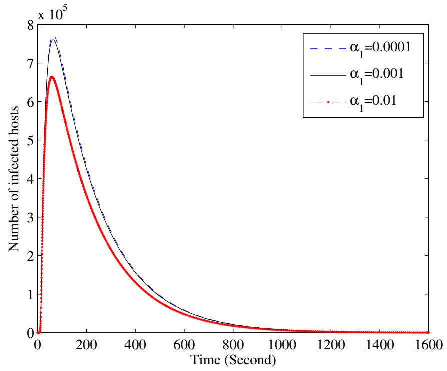


Figure 4: Effect of quarantined rate $\alpha_1$

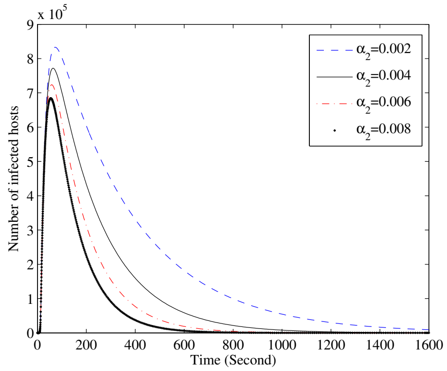


Figure 5: Effect of quarantined rate $\alpha_2$

## 6 Conclusions

This paper proposed an epidemic model to defend the propagation of active scanning worms, which takes the quarantine strategy into account. Firstly, we obtain the basic reproduction number using the next generation matrix. Next, with the help of the reproduction number, we prove the stabilities of worm-free equilibrium and endemic equilibrium. When the reproduction number is less than or equal to one, our model has only a worm-free equilibrium which is globally stable, which implies the worm dies out eventually; when the reproduction number is larger than one, our model has a unique endemic equilibrium which is globally stable, it implies that the worm persists in the whole population and tends to a steady state. Finally, simulation results are given to verify our conclusions. Our future work will expand this model which can characterize more features of Internet worms, e.g., taking delay or impulse into consideration.

## Acknowledgments

The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. This study was supported by the National Science Foundation of China under no. 61272541, China Postdoctoral Science Foundation of China under No. 2013M532018, and Foundation of Hebei Normal University under No. L2010B21.

## References

[1] R. Bellman, "Stability theory of differential equations," *Courier Dover Publications*, New York, 2013.

[2] Van Den Driessche, and J. Watmough, "Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission," *Mathematical Biosciences*, vol. 180, no. 1, pp. 29-48, 2002.

[3] H. I. Freedman, M. X. Tang, and S. G. Run, "Uniform persistence and flows near a closed positively invariant set," *Journal of Dynamics and Differential Equations*, vol. 6, no. 4, pp. 583-600, 1994.

[4] X. Han, and Q. Tan, "Dynamical behavior of computer virus on internet," *Applied Mathematics and Computation*, vol. 217, no. 6, pp. 2520-2526, 2010.

[5] J. P. LaSalle, "The Stability of Dynamical Systems, Regional Conference Series in Applied Mathematics," *SIAM, Philadelphia*, PA, 1976.

[6] M. Y. Li, J. R. Graef, L. C. Wang, and J. Karsai, "Global dynamics of an SEIR model with varying total population size," *Mathematical Biosciences*, vol. 160, no. 2, pp. 191-213, 1999.

[7] M. Y. Li, and J. S. Muldowney, "A geometric approach to global-stability problems," *SIAM. J. Math. Anal.*, vol. 27, no. 4, pp. 1070-1083, 1996.

[8] B. K. Mishra, and N. Jha, "SEIQRS model for the transmission of malicious objects in computer network," *Applied Mathematical Modelling*, vol. 34, no. 3, pp. 710-715, 2010.

[9] B. K. Mishra, and S. K. Pandey, "Fuzzy epidemic model for the transmission of worms in computer network," *Nonlinear Analysis: Real World Applications*, vol. 11, no. 5, pp. 4335-4341, 2010.

[10] B. K. Mishra, and S. K. Pandey, "Dynamic model of worms with vertical transmission in computer network," *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8438-8445, 2011.

[11] B. K. Mishra, and D. K. Saini, "SEIRS epidemic model with delay for transmission of malicious objects in computer network," *Applied Mathematics Computation*, vol. 188, no. 2, pp. 1476-1482, 2007.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, "Inside the slammer worm," *IEEE Security and Privacy*, vol. 1, no. 4, pp. 33-39, 2003.

[13] J. Ren, Y. Xu, Y. Zhang, Y. Dong, and G. Hao, "Dynamics of a delay-varying computer virus propagation model," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 371792, pp. 1-12, 2012.

[14] J. Ren, X. Yang, L. Yang, Y. Xu, and F. Yang, "A delayed computer virus propagation model and its dynamics," *Chaos Solitons Fractals*, vol. 45, no. 1, pp. 74-79, 2012.

[15] J. Ren, X. Yang, Q. Zhu, L. Yang, and C. Zhang, "A novel computer virus model and its dynamics," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 1, pp. 376-384, 2012.

[16] K. Simkhada, T. Taleb, Y. Waizumi, A. Jamalipour, and Y. Nemoto, "Combating against internet worms in large-scale network: an autonomic signature-based solution," *Security and Communication Networks*, vol. 2, no. 1, pp. 11-28, 2009.

[17] O. A. Toutonji, S. M. Yoo, and M. Park, "Stability analysis of VEISV propagation modeling for network worm attack," *Applied Mathematical Modelling*, vol. 36, no. 6, pp. 2751-2761, 2012.

[18] F. Wang, Y. Zhang, C. Wang, J. Ma, and S. J. Moon, "Stability analysis of a SEIQV epidemic model for rapid spreading worms," *Computers and Security*, vol. 29, no. 4, pp. 410-418, 2010.

[19] Y. Wei, W. Xun, C. Adam, X. Dong, and L. David, "On detecting active worms with varying scan rate," *Computer Communications*, vol. 34, no. 11, pp. 1269-1282, 2011.

[20] L. Yang, X. Yang, L. Wen, and J. Liu, "A novel computer virus propagation model and its dynamics," *International Journal of Computer Mathematics*, vol. 89, no. 17, pp. 2307-2314, 2012.

[21] L. Yang, X. Yang, Q. Zhu, and L. Wen, "A computer virus model with graded cure rates," *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 414-422, 2013.

[22] M. Yang, Z. Zhang, Q. Li, and G. Zhang, "An SLBRS model with vertical transmission of Computer virus over the Internet," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 925648, pp. 1-17, 2012.

[23] X. Yang, and L. Yang, "Towards the epidemiological modeling of computer viruses," *Discrete Dynamics in Nature and Society*, vol. 2012 Article ID 259671, pp. 1-11, 2012.

[24] Y. Yao, L. Guo, H. Guo, G. Yu, F. Gao, and X. Tong, "Pulse quarantine strategy of internet worm propagation: modeling and analysis," *Computers and Electrical Engineering*, vol. 38, no. 5, pp. 1047-1061, 2012.

[25] Y. Yao, X. Xie, H. Gao, Y. Ge, F. Gao, and X. Tong, "Hopf bifurcation in an Internet worm propagation model with time delay in quarantine," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2635-2646, 2013.

**Fangwei Wang** received his B.S. degree in 2000 from College of Mathematics & Information Sciences, Hebei Normal University, his M.S. degree in 2003 from College of Computer Science and Software, Hebei University of Technologyis, his Ph.D degree in 2009 from College of Computer at Xidian University. Currently he is an associate professor at Hebei Normal University, Shijiazhuang, China. His research interests include: network and information security, sensor networks.

**Yong Yang** received his B.S. degree in 1998 from Department of Information and Electronic Science, Yunnan University, his M.S. degree in 2003 from School of Information Science and Engineering, Yunnan University. Currently he is a lecturer at Yunnan University, Kunming, China. His research interests include network and information security.

**Yunkai Zhang** received his B.S. degree in 1986 from Department of Electronic and Information Engineering, Hebei University, his M.S. degree in 1997 from Department of Telecommunication Engineering, Beijing University of Posts and Telecommunications, and his Ph.D degree in 2005 from College of Computer at Xidian University. Currently he is a professor at Hebei Normal University, Shijiazhuang, China. His research interests include network and information security.

**Jianfeng Ma** received his B.S. degree in mathematics from Shaanxi Normal University, China in 1985, and obtained his M.E. and Ph.D. degrees in computer software and communications engineering from Xidian University, China in 1988 and 1995, respectively. Currently he is a Professor and Ph.D. supervisor in the School of Computer Science at Xidian University, Xian, China. His current research interests include distributed systems, wireless and mobile computing systems, computer networks, and information and network security.

# The Secure Transaction Protocol in NFC Card Emulation Mode

Yi-Lun Chi[1,2], Iuon-Chang Lin[3,4], Cheng-Hao Chen[3], and Min-Shiang Hwang[1,5]
*(Corresponding author: Iuon-Chang Lin)*

Department of Computer Science and Information Engineering, Asia University[1]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
Department of Marketing and Supply Chain Management, Overseas Chinese University[2]
100, Chiao Kwang Rd., Taichung 40721, Taiwan
Department of Management Information Systems, National Chung Hsing University[3]
250 Kuo-Kuang Rd., Taichung 402, Taiwan
(Email: corresponding_iclin@nchu.edu.tw)
Department of Photonics and Communication Engineering, Asia University[4]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University[5]

*(Received Feb. 15, 2015; revised and accepted Mar. 26 & Apr. 21, 2015)*

## Abstract

The NFC wallet is more popular in the world than before. Many people want to have a convenient payment in shopping, so the NFC wallet becomes an excellent choice for them. The NFC wallet uses the Card Emulation mode to achieve the transaction process. However, the Card Emulation mode does not have the specified secure transaction protocol. Our research provides a secure transaction protocol for the Card Emulation mode. It applies the Diffie-Hellman Key Exchange method and Elliptic Curve Cryptosystem to the protocol. It not only fulfills five secure requirements which are Data Confidentiality, Data Integrity, Unobservability, Unlinkability and Traceability, but also has the less calculation size and amount of transference than another proposed method. It is more suitable for mobile devices which do not have high calculation ability and storage space.

*Keywords: Elliptic curve cryptosystem, NFC security, secure transaction protocol*

## 1 Introduction

The mobile technology is more common than before, many people have smart phone or tablet for daily using. Near Field Communication (NFC) is a popular technology for payment which let users do not take their wallet out. It can establish a connected channel with touching. This behavior is simple and easy for use. The NFC transaction distance is in the 4 center meters. It is suitable for payment which reduces the risk of eavesdropping when transaction is being. The convenience make it been apply to many areas. Users usually use the NFC technology for transaction in the open environment [2]. Although every process is finished in few seconds, it still has many threats. Most importantly, the Peer-to-Peer mode has the secure protection protocol only, but the Card Emulation mode and Reader/Writer mode do not have any specified secure protocol in transaction. These two modes need mobile phone manufactures to design protection protocol when using mobile device in payment. Or users have to be careful for transaction object.

Our research considers it still have to take a secure protocol to protect private messages would not reveal when data exchanges. There are four international standards be introduced in next section. They are the base of NFC technology. In that part, our research describes the three operation modes for using in the NFC. We suppose the secure transaction protocol in the third partition. Our protocol uses the Diffie-Hellman Key Exchange method [7, 10, 11] and Elliptic Curve Cryptosystem [1] to build a key for exchanging the information between both sides. There is a comparison of our method and Hasoo et al. method [3] in Section 4. In addition, there is a security analysis about our protocol. In the conclusion, we provide some directions for research in future.

## 2 NFC Protocols and Operation Modes

The International Standard Organization (ISO) and International Electrotechnical Commission (IEC) have

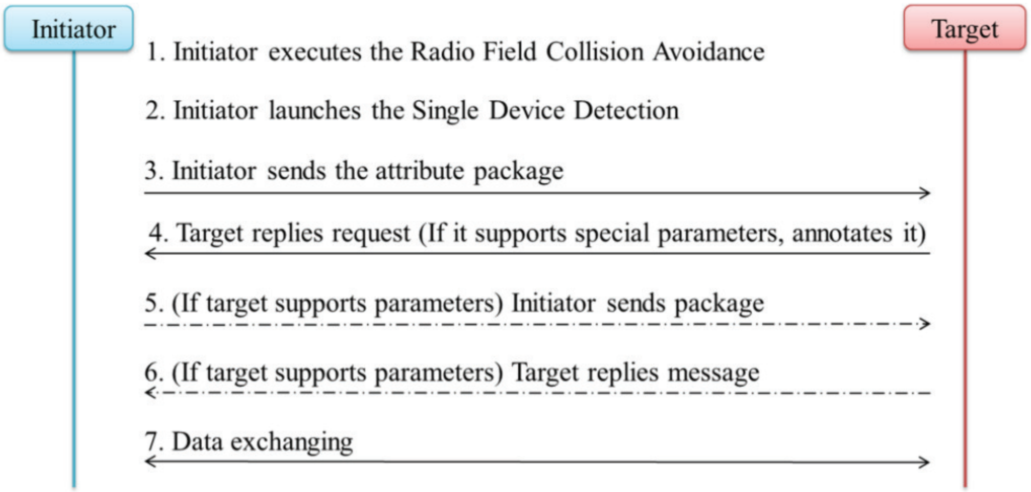

Initiator
Target
1. Initiator executes the Radio Field Collision Avoidance
2. Initiator launches the Single Device Detection
3. Initiator sends the attribute package
4. Target replies request (If it supports special parameters, annotates it)
5. (If target supports parameters) Initiator sends package
6. (If target supports parameters) Target replies message
7. Data exchanging

Figure 1: ISO/IEC 18092 passive communication mode

specified many standards for NFC communication. The ISO/IEC 18092 is the most important standard of NFC. The European Computer Manufacturers Association (ECMA) also defined the NFC security services and a protocol in ECMA International 385 standard and ECMA International 386 standard. In addition, NFC combines the smart card technology becomes a popular application. It makes mobile phone will be a tool which can pay the bill by NFC function. The detail specification has written in the ISO/IEC 14443 standard. In this section, here introduces the four international standards and three NFC operation modes.

## 2.1 ISO/IEC 18092 Standard

This standard defines the communication distance of NFC in the 4 centimeters. It make two devices can exchange data in short range. This feature elevates its security more for transaction. It also specified the two roles of objects which named "Initiator" and "Target" [4]. The initiator should launch a radio field for transaction at first and the target would get the radio waves from the initiator. If the target activates its radio field to communication with initiator, the mode names "Active Communication Mode", or it is called "Passive Communication Mode." The standard specified the NFC operating at center frequency is 13.56MHz. The transfer speeds are 106 kbit/s, 212 kbit/s and 424 kbit/s. The Figure 1 shows the detail procedure of initial process of Passive Communication Mode.

**Step 1.** When the initiator want to perform a transaction with target, it needs to execute the Radio Field Collision Avoidance (RFCA). It could detect there is existing the NFC radio field. If NFC field existed, this initiator should wait for other radio field disappeared after activates its radio field. If there are two radio fields operating in the same time, it may occur the data collision. If there is not, the initiator executes following steps.

**Step 2.** The initiator chooses the only one target for doing transaction. It decides the transfer speed and activates a radio field before using the Single Device Detection (SDD) to choose the target. For example, there are five targets in the area of radio field of initiator. The initiator should launch a polling request. The account of time slots is the data of the polling request. The time slot number is hexadecimal number which is '00', '01', '03', '07' or '0F' and it should be added 1. It means that the time slots number may be the one, two, four, eight or sixteen. As Figure 2, five targets have chosen the random number by themselves. When the delay time finished, they responds their numbers which match the number of time slots. If one slot has two responses of different targets that means there occurs a responding collision. According the rule, the initiator would decrypt the messages which sent by the target 1, target 2, target 4 and target 5. The initiator would find the target for transaction by their messages.

**Step 3.** The transaction target should be confirmed in Step 2. The initiator sends the attribute request to the target. Because it wants to know the ID of target and what transfer speed which target could support.

**Step 4.** Target would reply content of the request. If the target supports other special parameters or wants to adjust some current settings, it writes the requests down in the package. An instance is the target wants to enhance the transfer speed to higher rate, and it would write the need to the message and send to the target.

**Step 5.** If the initiator finds the target supports the special parameters by decrypting the message from the

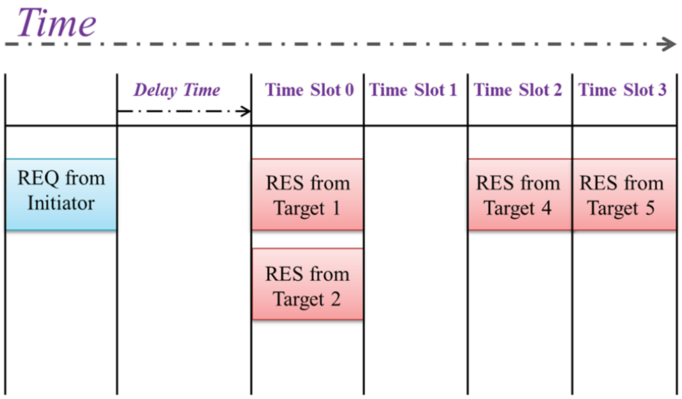


Figure 2: Single device detection method

target, it sends the request package to target. Or the initiator executes the Step 7 directly.

**Step 6.** The target receives the request and replies it. The content of package includes the adjusted items which the target wants. For example, the target wants to use higher speed for transaction that it should adjust the transfer rate from the 106 kbit/s to 212 kbit/s or 424 kbit/s.

**Step 7.** The initial process has finished that two sides would execute data exchanging procedure. Basically, all NFC transactions will do above steps before they start the data exchanging. However, if the target doesn't support the special parameters or does not want to change the current situation, it will not send the request to the initiator in the Step 4. So, the Step 5 and Step 6 are not must have be executed in every time.

## 2.2 ECMA International 385 and 386 Standards

The two international standards are specified by The European Computer Manufacturers Association (ECMA). They are used in the Peer-to-Peer mode of NFC operating mode. The two services of them which are shared secret service (SSE) and secure channel service (SCH) [8, 9]. The structure of ECMA international 385 standard describes the three layers as Figure 3. It divided to NFC-SEC User, NFC-SEC and NFC. When the user wants to contact to another one, its request would be written in the NFC-SEC-SDU (Service Data Unit). The NFC-SEC-SAP (Service Accessing Point) will invoke the communication service. And it combines the NFC-SEC-SDU with the NFC-SEC-PCI (Protocol Control Information) to the NFC-SEC-PDU (Protocol Data Unit). After the NFC-SEC-SAP contacts with the NFC-SAP (Service Accessing Point), it sends the NFC-SEC-PDU to the NFC-SAP. The NFC-SAP will send the communication request to another NFC-SAP for establishing a connection. This process makes them to coordinate the shared secret key to protect the privacy. It is called shared Secret service (SSE) and secure channel service (SCH). The implement method is described in the ECMA International 386 standard. The realized process is using the Elliptic Curve Cryptosystem and the Diffie-Hellman Key Exchange method to generate the secret key for transaction.

## 2.3 ISO/IEC 14443 Standard

It defines the transaction protocol which smart card communicates with card reader and specifies the 13.56MHz is the main radio frequency. It originally consists of two transmission technologies which are NFC-A and NFC-B. But the SONY Corporation wants to make their technology “FeliCa” been combined to this international standard, so SONY had written a draft and sent to related organization for judging. The International Standard Organization judged the draft is fail, because the specified content is similar the ISO/IEC 18092. However, the non-profit organization for promoting the NFC technology which named NFC Forum, it still make the three transmission technologies “ISO/IEC 14443A”, “ISO/IEC 14443B”, and “SONY FeliCa” to be named “NFC-A”, “NFC-B” and “NFC-F”. The mission of NFC Forum is formed to advance the NFC by developing related specifications and ensuring the interoperability among devices. The three technologies are divided by their encoded modes, Modulation Types and data rates. As the Figure 4, there is an initial process before exchanging data between two devices [5, 6]. Here introduces the detail about the procedures.

**Step 1.** The mobile phone which embedded the smart card or secure chip should touch the reader device.

**Step 2.** The smart card (It represents that mobile phone) waits for a wake-up command for executing following process for this transaction.

**Step 3.** Reader sends the wake-up commands to mobile phone.

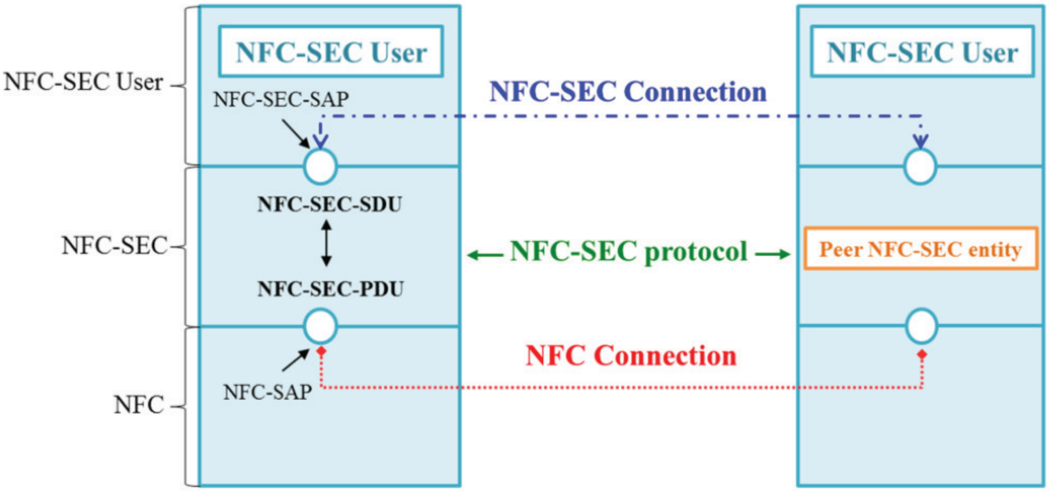


Figure 3: NFC-SEC architecture

**Step 4.** Smart card replies the request from the reader. If smart card supports some special parameters or like to change current settings, it should write requirements down in this message and send to reader.

**Step 5.** In order to avoid the data collision, the reader activates the anti-collision protocol. It would detect does there have other smart card in the radio field. If there only one smart card in the field, the reader executes the next step.

**Step 6 & step 7.** Because the target has confirmed, the reader sends the request to smart card for getting the unique ID of smart card. However, the unique ID had divided three parts by smart card. It is used to check the card's specification that conforms to the ISO/IEC 14443 standard. So they execute these two steps three times for getting complete ID and inspecting the specification of card is eligible.

**Step 8.** If the smart card annotated the special parameters in the message of step 4, the reader should send the attribute package to card. Otherwise, the Step 8 and Step 9 do not be executed.

**Step 9.** If card received the package from the reader, it fills in the information which it needs in this transaction and sends it back. The reader should adjust the current settings according by the message.

**Step 10.** If all situation is right, they start to exchange data.

## 3 The Secure Transaction Protocol in NFC Card Emulation Mode

In the card emulation mode, it does not have a transaction protocol which like the ECMA International 385 and ECMA International 386 standards. The transaction security of card emulation mode depends on the defense of software or hardware by the manufactures. We got the inspiration from those two standards. Our method uses the Diffie-Hellman Key Exchange method and Elliptic Curve Cryptosystem (ECC) in the card emulation mode. It describes our method as following.

Our method is established by Elliptic Curve Cryptosystem and Diffie-Hellman Key Exchange method. It helps two sides of transaction can obtain a same session key for transmission. They can use the key to encrypt their information when exchange secret data. If the key would not easily to be cracked, the transaction process should be more safety. First, the user's smart card and the NFC reader of store have to know their ID by each other. Then, they choose a huge prime number $p$. The elliptic curve $y^2 = x^3 + ax + b$ should in the finite field of $Z_P$. The a,b $\in Z_p$ and a,b $< p$. The a,b also should fulfill the condition which $4a^3 + 27b^2 \neq 0$. The elliptic curve which is $y^2 = x^3 + ax + b \bmod\ p$ can be used to encrypt and decrypt. The smart card and reader choose the base point from this curve which named $G$. It would be used to generate the public key. When all of above has done, it would execute the protocol. The notation follows Table 1.

As follow Figure 5 to Figure 8, it includes ten steps for executing the transaction in our protocol. It describes the steps detailed in this section. The Steps 1 and 2, there are in the Figure 5.

**Step 1.** The smart card generates a random number $x$ , it uses the point multiplication with base point $G$ for

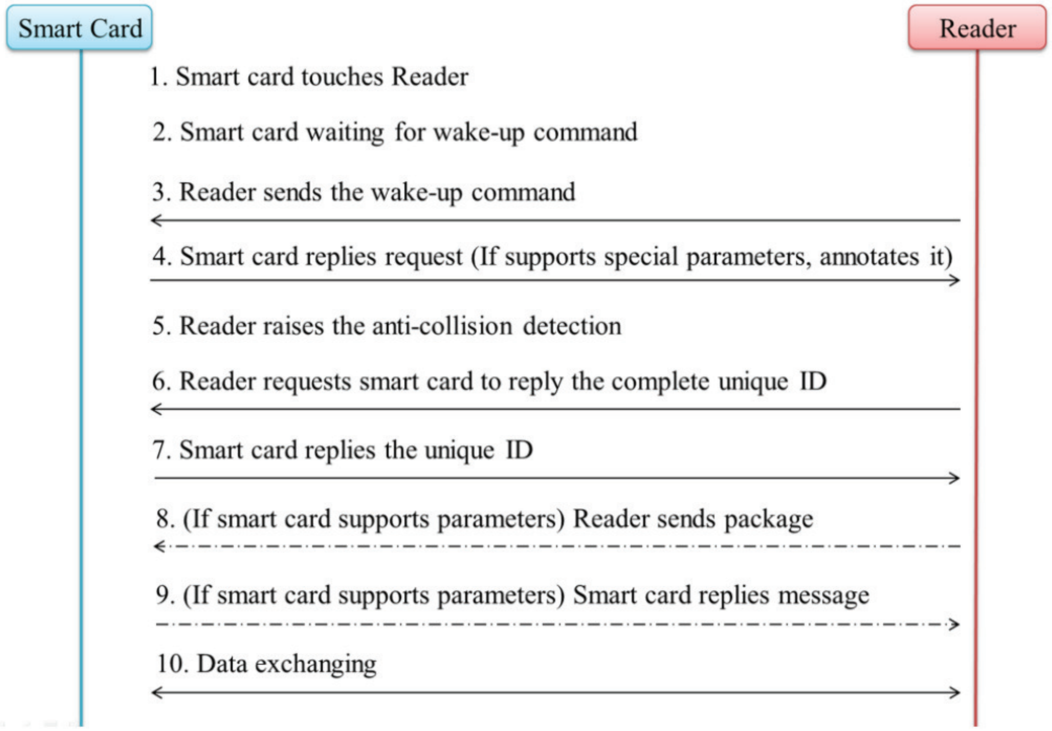

Smart Card
Reader
1. Smart card touches Reader
2. Smart card waiting for wake-up command
3. Reader sends the wake-up command
4. Smart card replies request (If supports special parameters, annotates it)
5. Reader raises the anti-collision detection
6. Reader requests smart card to reply the complete unique ID
7. Smart card replies the unique ID
8. (If smart card supports parameters) Reader sends package
9. (If smart card supports parameters) Smart card replies message
10. Data exchanging

Figure 4: ISO/IEC 14443 communication mode

Table 1: Notations

| Symbol | Description |
|---:|---|
| $x$, $y$ | Random numbers |
| $Q_x$, $Q_y$ | Public keys |
| $s$ | Secret number |
| $K$ | Session key |
| $fc$ | Function of AES in XCBC-PRF-128 mode |
| $ID_x, ID_y$ | Random ID value |
| $MacTag_x$<br>$MacTag_y$ | Key certificated tags |
| $EL$ | Encrypted shopping list |
| $l$ | Shopping list |
| $TP_x, TP_y$ | Total price |
| $P_t$ | Price table |
| $TPA$ | Total price and APDU commands |
| $APDU$ | Application protocol data unit |
| $ER$ | Enrypted result |
| $R$ | Result of executing |
| $stc$ | Credit card's three secure codes |
| $pK$ | Public key of bank |
| $phone$ | User's phone number |

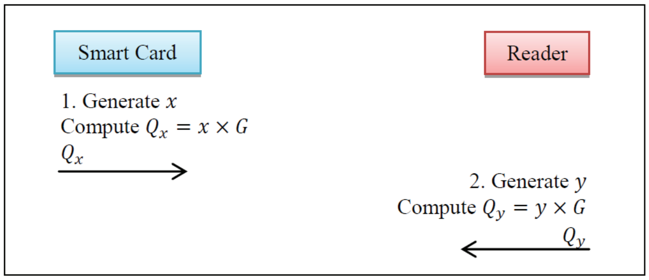


Figure 5: Exchange the key each other

generating the public key of card which is $Qx$. The card sends the key to reader.

**Step 2.** After the reader receives the message, it would make a random number which is $y$. Reader uses the same method to generate public key $Qy$ and sends it to smart card.

**Step 3.** After card had received $Qy$ from reader, it took the random number $x$ and $Qy$ to do point multiplication of elliptic curve. It got the point $P$ of $y^2 = x^3 + ax + b \bmod p$ and set the value of $x$-way which named $s$. The smart card used the AES-XCBC-PRF-128 algorithm to encrypt ($Qx$,$Qy$,$s$), and got the key $K$. As the same time, the reader got $Qx$ from smart card, it took $Qx$ do the point multiplication by random number $y$. It can take the value $s$ from the

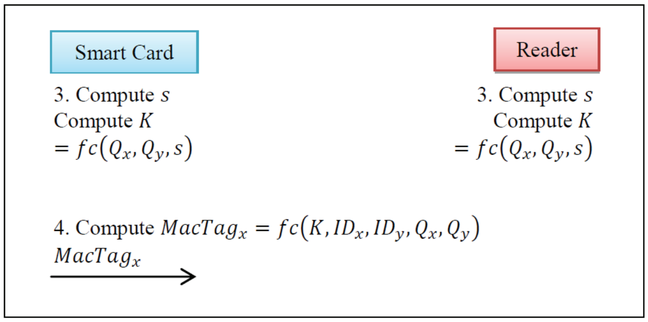


Figure 6: Generate the session key and certificated tag

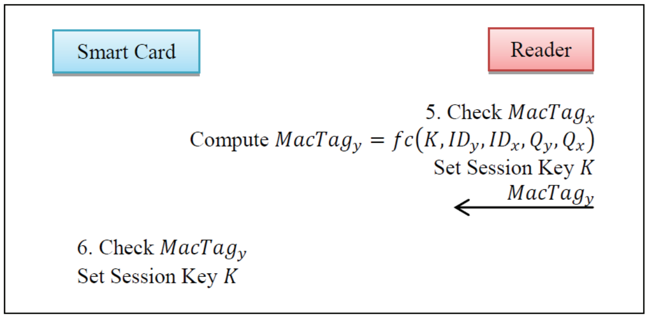


Figure 7: Set session key process

$x$-way which the point $P$ of $y^2 = x^3 + ax + b \bmod p$. The reader device used the AES-XCBC-PRF-128 algorithm to encrypt ($Qx$,$Qy$,$s$), and would gain the same key $K$ as smart card.

**Step 4.** The smart card encrypted five items which are session key $K$, two random identified value $IDx$ and $IDy$, and two public keys $Qx$ and $Qy$ to the $MacTag_x$ by AES-XCBC-PRF-128 algorithm. It sent the tag to the card reader for checking the identity. The AES-XCBC-PRF-128 algorithm is a design on Cipher Block Chaining Message Authentication Code Calculation. This method would divide a data to many blocks. The 1 to $n-1$ blocks have same length. The length is uniquely 128 bits. The length of final block is in the 1 bit to 128 bits. If one user takes his data for encrypting by this algorithm, he can send the ciphertext to another user. That receiver would check the result which using own data to encrypt by the algorithm is identical or not. It could identify their data is totally same or not at all.

**Step 5.** When reader had received $MacTag_x$, it would identify the validity. If pass, card reader encrypted the key $K$, their identification $IDy$ and $IDx$, their public keys $Qx$ and $Qy$ to the $MacTag_y$ by AES-XCBC-PRF-128 algorithm. The reader has known the key $K$ which they own are same, and it set the key as the session key for this transaction. Reader sent the $MacTag_y$ to smart card.

**Step 6.** The card identified the content of $MacTag_y$. If correct, the smart card set the key $K$ as the session key.

**Step 7.** The card used the session key $K$ to encrypt user's shopping list and sent encrypted shopping list $EL$ to card reader of store.

**Step 8.** After reader received the data, it decrypted the list by the session key $K$. It checked each items' price and calculated the total price of customer should pay as $TP_y$. Reader took the key $K$ to encrypt the $TP_y$ and the APDU commands became a data which was named $TPA$. It sent the $TPA$ to smart card.

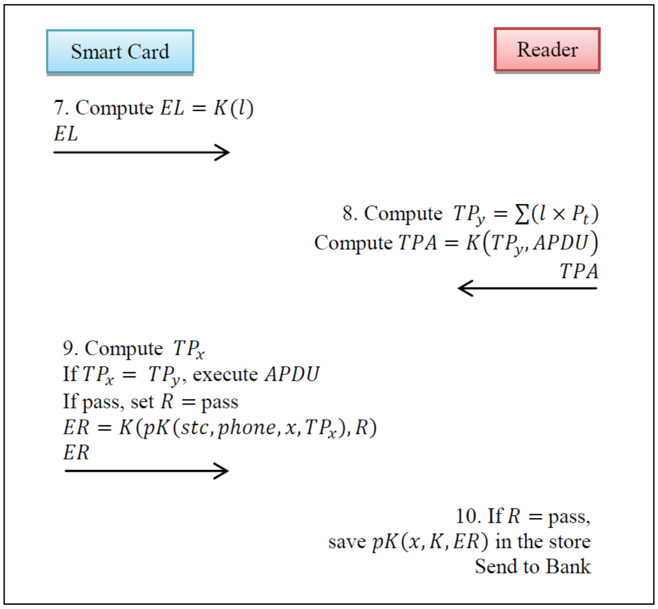

Smart Card
Reader
7. Compute $EL = K(l)$
$EL$
8. Compute $TP_y = \sum(l \times P_t)$
Compute $TPA = K(TP_y, APDU)$
$TPA$
9. Compute $TP_x$
If $TP_x = TP_y$, execute $APDU$
If pass, set $R$ = pass
$ER = K(pK(stc, phone, x, TP_x), R)$
$ER$
10. If $R$ = pass,
save $pK(x, K, ER)$ in the store
Send to Bank

Figure 8: The process of message transportation

**Step 9.** Card calculated the total price first, it has named $TP_x$. If $TP_x$ equals to $TP_y$, the smart card would execute APDU commands. If the process was smoothly and successfully, card created the result of executing which was named $R$. The content of $R$ should be set pass. The smart card uses the bank's public key $pK$ to encrypt the Credit card's three secure codes, user's phone number, random number $x$ and $TP_x$. It took this data and R has been encrypted by key $K$. The card sent the encrypted result $ER$ to reader.

**Step 10.** If checked result of $R$ was pass, the reader would use bank's public key $pK$ to encrypt the random number $x$, session key K and encrypted result $ER$. This ciphertext should be stored into database of the store. Finally, it would be sent to bank for follow-up process.

# 4 Performance and Security Analysis

## 4.1 Comparison with Hasoo et al.'s Method

Hasoo et al. has proposed a communication method on NFC card emulated mode [3]. It also used the ECMA International 385 Standard and ECMA International 386 Standard to build the procedure. However, the method of Hasoo et al. has described the process of transaction until two sides had agreed on session key K. It equals as our method of step 1 to 6. This article would take the data calculation and the amount of transmission for comparison with Hasoo et al. method. These two methods has been proposed secure method on Card Emulated Mode, therefore our comparison aim at smart card exclusive of the reader side. The result shown as Table 2.

Table 2: The comparison of two methods

| Methods | Data Calculation | Amount of Transmission |
|---|---|---|
| Ours | 2432 bits | 592 bits |
| Hasoo et al. | 1728 bits | 480 bits |

## 4.2 Security Analysis of Protocol

Our protocol not only has less calculation than Hasoo et al. method, but also satisfies five secure requirements of NFC transaction which proposed by Hasoo et al. Here, we would describe the five items as follows.

### 4.2.1 Data Confidentiality

"Data Confidentiality" means the data would not be accessed, traced or analyzed by unauthorized user. In our protocol, if a smart card wants to communicate with reader, and it would uses the Elliptic Curve Cryptosystem and Diffie-Hellman Key Exchange method to make a session key. If anyone likes to fake this key, he has to solve the discrete logarithm problem first. It is still a very difficult thing now. On the other hand, it can protect data would not be used by unauthorized user in our protocol.

### 4.2.2 Data Integrity

"Data Integrity" represents the data should keep its integrity when transmitting, and it would not be adjusted or edited by illegal one. As the ISO/IEC 18092 standard describes "Any device should activate the Radio Field Collision Avoidance (RFCA) and the Single Device Detection (SDD) before communicating with others." It promises the target would be only one until finish that process. In other words, there does not have third party could steal, adjust or edit the data before transaction completed. So our protocol satisfies this requirement.

### 4.2.3 Unobservability

"Unobservability" describes that any unauthorized user can not find the specified user's data by observing or analyzing. Our method proposes the smart card should take a random number be the base number of smart card's public key. When the transaction has finished, the key would be abandoned. If illegal man intruded the database of store, he could not observe the specified user's identification.

### 4.2.4 Unlinkability

"Unlinkability" is which two data had generated by same user, and it does not have any linkability between them. In our protocol, even a user has many times for shopping in same store, his transaction identification is different every time. Because the unique ID of smart card and the base number of user's public key were randomly selected out, it makes others can not find the relationship of every shopping record in the database of store.

### 4.2.5 Traceability

"Traceability" defines if there was a transaction problem occurred, it has a property for investigating the truth in the dispute between two sides. This character is usually held by a third party. Due to there has a data of transaction being stored in the store's database in our method, it would help to investigate the truth. The third party can decrypt the data and get the import items just like the user's phone number, the original random number which was the base of user's public key and final transaction result which had authorized by user. They would help to solve the problem.

# 5 Conclusions

Our research has found there is not a standardized secure transaction protocol for Card Emulation Mode, so this article uses the Diffie-Hellman Key Exchange method and Elliptic Curve Cryptosystem to build a secure communication protocol for the mode. It fulfills five secure transaction requirements which are unobservability, data integrity, data confidentiality, unlinkability, and traceability. And the data calculation and the amount of transmission are better than Hasoo et al. method. Hence, our protocol has nice performance in security and calculation.

# References

[1] S. Basu, "A new parallel window-based implementation of the elliptic curve point multiplication in multi-core architectures," *International Journal of Network Security*, vol. 14, no. 2, pp. 101–108, 2012.

[2] A. K. Das, "Improving identity-based random key establishment scheme for large-scale hierarchical wireless sensor networks," *International Journal of Network Security*, vol. 14, no. 1, pp. 1–21, 2012.

[3] E. Hasoo, "Conditional privacy preserving security protocol for nfc applications," *Proc. 2013 IEEE Int. Conf. On Consumer Electronics*, 2013.

[4] ISO/IEC 18092:2013, *Information Technology - Telecommunications and Information Exchange between Systems - Near Field Communication - Interface and Protocol (NFCIP-1),* ISO/IEC 18092:2013.

[5] ISO/IEC FCD 14443-3, *Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part 3: Initialization and Anti-collision,* ISO/IEC FCD 14443-3 (Revision).

[6] ISO/IEC FCD 14443-4, *Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part 4: Transmission Protocol,* ISO/IEC FCD 14443-4 (Revision).

[7] J. Liu and J. Li, "A better improvement on the integrated Diffie-Hellman-DSA key agreement protocol," *International Journal of Network Security*, vol. 11, no. 2, pp. 114–117, 2010.

[8] Standards ECMA 385 NFC-SEC, *NFCIP-1 Security Services and Protocol,* Standards ECMA 385 NFC-SEC.

[9] Standards ECMA 386 NFC-SEC-01, *NFC-Sec Cryptography Standard Using ECDH and AES,* Standards ECMA 386 NFC-SEC-01.

[10] S. Wu and Y. Zhu, "Proof of forward security for password-based authenticated key exchange," *International Journal of Network Security*, vol. 7, no. 3, pp. 335–341, 2008.

[11] Z. Yong, Ma Jianfeng, and S. Moon, "An improvement on a three-party password-based key exchange protocol using weil pairing," *International Journal of Network Security*, vol. 11, no. 1, pp. 14–19, 2010.

**Yi-Lun Chi** received her M.S. degrees in Management of Information Systems and Technology from School of Information Systems and Technology, Claremont Graduate University, USA in 2006 and in Computer Science from University of Southern California in 1997. She is currently being an instructor at Overseas Chinese University and pursuing the Ph.D. degree in the department of Computer Science and Information engineering at Asia University. Her research interests include electronic commerce, internet marketing, data mining, and knowledge management.

**Iuon-Chang Lin** received the Ph.D. in Computer Science and Information Engineering in March 2004 from National Chung Cheng University, Chiayi, Taiwan. He is currently a professor of the Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan. His current research interests include electronic commerce, information security, RFID Information Systems, and cloud computing.

**Cheng-Hao Chen** received the M.S. in Management Information Systems from Chung Hsing University, Taiwan, in 2014; His current research interests include mobile agent, information security, and cryptography.

**Min-Shiang Hwang** received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications.

# Repairable Image Authentication Scheme

Yi-Hui Chen[1,2], Chih-Yang Lin[1,2], Wanutchaporn Sirakriengkrai[1], and I-Chun Weng[1]
(Corresponding author: Chih-Yang Lin)

Department of Computer Science and Engineering, Asia University[1]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
(Email: andrewlin@asia.edu.tw)
Department of Medical Research, China Medical University Hospital, China Medical University[2]
Taichung 40402, Taiwan

(Received May 16, 2013; revised and accepted Jan. 24 & Apr. 8, 2014)

## Abstract

Nowadays, authentication mechanism is widely applied to digital images to verify whether the received image is not a fake. In this paper, we propose a self-authentication mechanism without any extra data to authenticate whether the area is modified by comparing the generated authentication code and hidden authentication code together. Also, the recovery ability is employed to the proposed scheme used to repair the modified area. In our experimental result, we show the positive result for the feasibility of the proposed scheme.

*Keywords: Image authentication, located mechanism, recovery*

## 1 Introduction

Image authentication is a mechanism to authenticate whether the areas is modified during transmission over the internet. The content of the image may be replaced with fake information, thus to identify the fake area even to recovery it after it is judged as an illegal place. One class of fragile watermarking method [2, 5, 10, 11, 12], the original image is separated into several small blocks; then, embeds the watermark into these blocks. While tempering the image, the matching between the content and the watermark in the corresponding block will be destroyed. As for pixel-wise fragile watermarking scheme [3, 4, 6, 7, 8, 9], this method the tampered pixels can be specified from the absence of the carried watermark. In other work [13], a hierarchical fragile watermarking mechanism, this method obtains the watermark data from pixels and blocks.

The receiver can identify the blocks inauthentic according to the watermark hidden in other blocks to locate the tampered pixels. The scheme [13] combined the advantages of block-wise and pixel-wise technique to find the detailed tampering pattern even though the modified area is too large. Some watermarking approach with content restoration is not feasible because the tampered area is too large to locate tampered pixels. In scheme [13], it has a limit that the tempered pixels cannot be restored if the percentage of tampered area is more than 6.6%. In scheme [1], the features of an image are obtained from the cryptographic hash function which only the owner can prove the rightful ownership with the pre-determined secret key. The scheme [1] can achieve the tampering detection for ownership protection, but it cannot recover the tampered areas.

In this paper, we proposed a self-authentication mechanism as well as recovery abilities for digital images. In the proposed scheme, the authentication codes for a digital image generated by itself with recovery data are to hides back into the original one. After that, receivers can extract the hidden data to check whether it is not a fake one. The fake area is detected and marked. Later on, the extracted recovery data could be used to repair the tampered area without any extra data.

## 2 Related Work

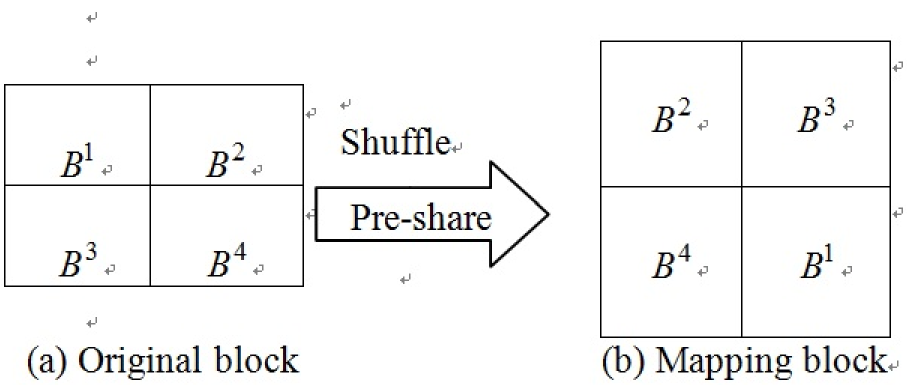
In this section, we introduce traditional (t, n)-threshold secret sharing.

Shamir *et al.* proposed the $(t, n)$-threshold secret sharing as shown in Equation (1), which the secret is treated as the parameter $r_1$ and the other parameters $r_2$, $r_3$, $\cdots$, $r_t$ are chosen by random to construct a $(t-1)$-degree polynomials.

$$R(x_i) = r_1 + r_2x_i + r_3x_i^2 + \cdots + r_tx_i^{t-1}, \tag{1}$$

where the value of $x_i$ is the ID of the $i^{th}$ participant, and all $x_i$'s are individual from each other. Hence, $n$ participants will construct $n$ $(t-1)$-degree polynomials, as shown in Equation (2).

$$\begin{cases} R(x_1) = r_1 + r_2x_1 + r_3x_1^2 + \cdots + r_tx_1^{t-1}, \\ \quad\quad\vdots \\ R(x_n) = r_1 + r_2x_n + r_3x_n^2 + \cdots + r_tx_n^{t-1} \end{cases} \tag{2}$$



(a) Original block (b) Mapping block

Figure 1: Illustration of a four-pixel block

The unknown messages $r_1$ can be resolved with the polynomial interpolation, shown in Equation (3), where $t$ participants join to re-constructing procedure. The $(t-1)$-degree polynomial will be reconstructed.

$$\begin{aligned} R(x) &= R(x_1)(\frac{x-x_2}{x_1-x_2})(\frac{x-x_3}{x_1-x_3})\cdots(\frac{x-x_t}{x_1-x_t})+ \\ &\vdots \\ & R(x_t)(\frac{x-x_1}{x_t-x_1})(\frac{x-x_2}{x_t-x_2})\cdots(\frac{x-x_{t-1}}{x_t-x_{t-1}}). \end{aligned} \tag{3}$$

# 3 The Proposed Scheme

The proposed scheme consists of four procedures:

1) The secret sharing procedure;

2) The authentication generation procedure;

3) The authentication;

4) Secret reconstruction procedures and recovery of the inauthentic area.

First, the secret sharing procedure shows how the secret image is separated into shares. Second, the authentication codes generation procedure shows how to generate the authentication code and the recovery data. Third, the authentication and reconstruction procedures prove whether the image is authentic. If not, the inauthentic area will be marked and adjust the results of the authentication procedure. After that, the inauthentic area will be repaired during the last procedure.

## 3.1 Secret Sharing Procedure

Assume that the original image is the size of $n \times n$ pixels, where $n = 4$. Then divides original image into non-overlapping block with $2 \times 2$ pixels and pixels of the $\mu$-th block are denoted as $P_1^\mu$, $P_2^\mu$, $P_3^\mu$ and $P_4^\mu$ where $\mu$ is the block ID, and $1 \leq \mu \leq \frac{n \times n}{16}$. The average value for the $\mu$-th block, denoted as $B^\mu$, i.e., $B^\mu = \sum_{i=1}^{4} P_i^\mu/4$. After that, a pre-shared key is used to shuffle the block positions shown in Figure 1(a) and 1(b) as the results before shuffled and after shuffled, respectively. For a given block the average value of mapping block is denoted as $\bar{B}^\mu$. $B^\mu$ and $\bar{B}^\mu$ are treated as a partner-block pair. For example, $B^1$ and $B^2$ are a partner-block pair in Figure 1. Next the value of is represented with a 7-based notation. For example, if $\bar{B}^\mu = 100$ and translated into 7-based notations as $(202)_7$, and the digits in $\bar{B}^\mu$ are denoted as $\bar{B_1}^\mu$, $\bar{B_2}^\mu$, and $\bar{B_3}^\mu$, respectively. That is, $\bar{B_1}^\mu = 2$, $\bar{B_2}^\mu = 0$, and $\bar{B_3}^\mu = 2$.

We reconstruct a formula with the values of $\bar{B_1}^\mu$, $\bar{B_2}^\mu$, and $\bar{B_3}^\mu$ as Equation (4).

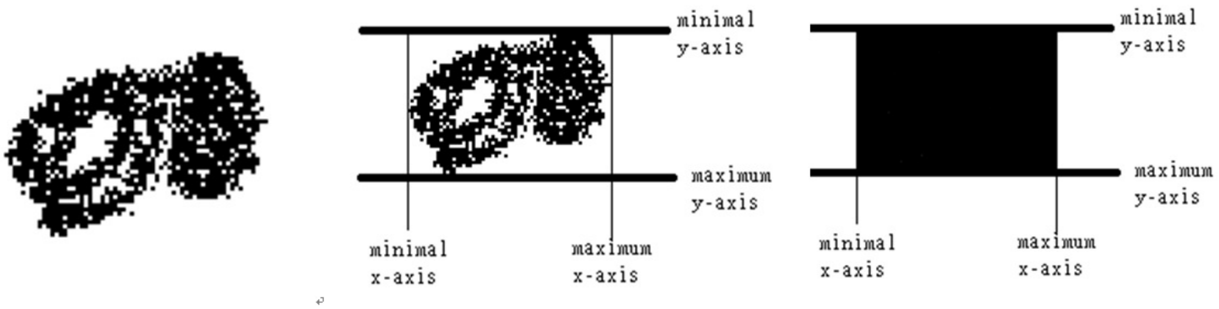
$$R_B(x_i) = \bar{B_1}^\mu + \bar{B_2}^\mu x_i + \bar{B_3}^\mu x_i^2 \bmod 7. \tag{4}$$

Here, the notation $i$ means the $i$-th input value and $\mu$ is the block ID. Assume that $x_1$, $x_2$, and $x_3$ are 2, 3, and 5, respectively. If $\bar{B_1}^\mu$, $\bar{B_2}^\mu$, and $\bar{B_3}^\mu$ are 2, 0, 2, the formula are built through Equation (1) as $R_B(x_i) = 2 + 0x_i + 2x_i^2 \bmod 7$. Then we input the predefined value of $x_1$, $x_2$, and $x_3$ to be as 2, 3, and 5 and input into Equation (1) to get the value of $R_B(x_1 = 2)$, $R_B(x_2 = 3)$, and $R_B(x_3 = 5)$ as 3, 6, and 3, respectively.

Later on, we translate $R_B(x_1 = 2)$, $R_B(x_2 = 3)$, and $R_B(x_3 = 5)$ into binary bit streams and depicted as $A_{b1}^\mu$, $A_{b2}^\mu$ and $A_{b3}^\mu$. For example, while the values of $R_B(x_1 = 2)$, $R_B(x_2 = 3)$, and $R_B(x_3 = 5)$ are 3, 6, and 3, the values of $A_{b1}^\mu$, $A_{b2}^\mu$ and $A_{b3}^\mu$ will be $(011)_2$, $(110)_2$, and $(011)_2$, respectively. After that, we embed the generated shares into original image by replacing the least three significant bits of pixels $P_1^\mu$, $P_2^\mu$ and $P_3^\mu$ with the values of $A_{b1}^\mu$, $A_{b2}^\mu$, $A_{b3}^\mu$, and $A_{b4}^\mu$ to generate the stego-pixels. After embedding, stego-pixel values are 163, 150 and 171, and depicted as $\bar{P_1}^\mu$, $\bar{P_2}^\mu$, $\bar{P_3}^\mu$ and $\bar{P_4}^\mu$.

## 3.2 The Authentication Generation Procedure

In this procedure, we describe how to generate the authentication codes for block $B^\mu$. The authentication



(a) Authentication detection results

(b) The area of authentication detection results

(c) Inauthentic area adjustment

Figure 2: Adjustment method of the authentication results

codes generated for the block $B^\mu$ is denoted as $A^\mu$. The given block $B^\mu$ contains four pixels, i.e., $P_1^\mu$, $P_2^\mu$, $P_3^\mu$ and $P_4^\mu$. Let four pixels be a group, and each pixel must be transformed into a binary stream. For example, when $P_1^\mu = 161$, $P_2^\mu = 146$, $P_3^\mu = 170$, and $P_4^\mu = 86$, then translated into a binary bit stream as $(10100000)_2$, $(10010010)_2$, $(10101010)_2$, and $(01010110)_2$, respectively. Later on, we keep the five most significant bits of $P_i^\mu$ and represented as $P_{i,1}^\mu$, $P_{i,2}^\mu$, $P_{i,3}^\mu$, $P_{i,4}^\mu$, and $P_{i,5}^\mu$, for $i = 1$ to 4. For example, the most five significant bits of $P_i^\mu$ are presented by $P_{1,1}^\mu$, $P_{1,2}^\mu$, $P_{1,3}^\mu$, $P_{1,4}^\mu$, and $P_{1,5}^\mu$ as 1, 0, 1, 0, and 0, respectively. The authentication must be generated with Equation (5) and the values of $A_1^\mu$, $A_2^\mu$, $A_3^\mu$, $A_4^\mu$, and $A_5^\mu$ must be generated with Equation (6), for $j = 1$ to 5, "$\oplus$" means $XOR$ (Exclusive OR) operation.

$$A^\mu = A_1^\mu \times 2^4 + A_2^\mu \times 2^3 + A_3^\mu \times 2^2 + A_4^\mu \times 2 + A_5^\mu \bmod 8. \tag{5}$$

$$A_j^\mu = P_{1,j}^\mu \oplus P_{2,j}^\mu \oplus P_{3,j}^\mu \oplus P_{4,j}^\mu. \tag{6}$$

For example, with Equation (6), $A_1^\mu = 1 \oplus 1 \oplus 1 \oplus 0 = 1$; and thus $A_1^\mu = 1$, $A_2^\mu = 1$, $A_3^\mu = 0$, $A_4^\mu = 0$, and $A_5^\mu = 1$, respectively. Through Equation (5), $A^\mu$ is equal to 1 (i.e., $A^\mu = 25 \bmod 8 = 1$). Then $A^\mu$ transform into a 3-bit binary stream, denoted as $A_{b4}^\mu = (001)_2$. Finally, we embed the generated authentication codes into the pixel $P_4^\mu$ by replacing the least three significant bits of pixel $P_4^\mu$ with the value of $A_{b4}^\mu$ to generate the stego-pixels. After embedding, the fourth pixel value of the block $B^\mu$ is changed as 81, which is depicted as $\bar{P_4}^\mu$.

## 3.3 Authentication Procedure

In this procedure, we describe the way to extract the hidden information and to authenticate whether the secret image is an authentic one. The watermarked image is the size of $n \times n$ pixels then divided into non-overlapping blocks with size of $2 \times 2$ pixels a block. For a given block, four pixels in the $\mu$-th block are denoted as $\bar{P_1}^\mu$, $\bar{P_2}^\mu$, $\bar{P_3}^\mu$, and $\bar{P_4}^\mu$, where $\mu$ is the block ID.

For a given pixel $\bar{P_i}^\mu$, we keep the five most significant bits and represented as $\bar{P_{i,1}}^\mu$, $\bar{P_{i,2}}^\mu$, $\bar{P_{i,3}}^\mu$, $\bar{P_{i,4}}^\mu$, and $\bar{P_{i,5}}^\mu$, for $i = 1$ to 4. For example, if $\bar{P_1}^\mu = 163$, the binary bits stream will be $(10100)_2$. Therefore, the most five significant bits are presented by $\bar{P_{i,1}}^\mu$, $\bar{P_{i,2}}^\mu$, $\bar{P_{i,3}}^\mu$, $\bar{P_{i,4}}^\mu$, and $\bar{P_{i,5}}^\mu$ as 1, 0, 1, 0, and 0, respectively. The authentication code must be generated with Equation (5). Then, we can extract the hidden authentication codes by the last three bits pixel of the block. We compare the generated authentication codes with the extracted authentication codes together to check whether they are the same. If true, it judged as an authentic block; otherwise, inauthentic.

There is a risk that some pixels are not detected as inauthentic pixel as shown in Figure 2(a). Thus, the biggest area covers all the possible inauthentic pixels to adjust the authentication results. That is, the most minimal and most maximum x-axis and y-axis are recorded. The pixels located at the range from the minimal x-axis to the maximum x-axis and also in the range of minimum y-axis and maximum y-axis are treated as inauthentic pixels. As shown in Figure 2, Figure 2(a) is the original authentication results. The minimal and maximum x-axis and y-axis are found as shown in Figure 2(b). All the pixels locate at the range of the minimal and maximum x-axis and y-axis are treated as inauthentic pixels as shown in Figure 2(c).

## 3.4 Reconstruction Procedure

The pixels $\bar{P_1}^\mu$, $\bar{P_2}^\mu$, $\bar{P_3}^\mu$, and $\bar{P_4}^\mu$ are transformed into a binary streams, individually. Next, we get the last three bits of the transformed binary stream and then transform the 3-bit binary stream into three decimal digits as the returned values of $\bar{R_B}(x_1 = 2)$, $\bar{R_B}(x_2 = 3)$, and $\bar{R_B}(x_3 = 5)$. Finally, the values of $\hat{B_1}^\mu$, $\hat{B_2}^\mu$, and $\hat{B_3}^\mu$ can be obtained with Equation (7). The mean value of the mapping block, denoted as $\hat{B}^\mu$ is obtained with Equa-

Table 1: Visual qualities of the five test images

| | Visual qualities | | |
|---|---|---|---|
| Images | PSNR of watermarked image | PSNR of modified Image | PSNR of recovery image |
| Barbara | 38.13 | 27.05 | 35.82 |
| Baboon | 38.58 | 25.03 | 39.09 |
| Boats | 38.13 | 29.26 | 30.06 |
| Cartoon | 37.71 | 23.34 | 30.12 |
| Goldhill | 38.13 | 25.92 | 30.20 |

tion (8).

$$
\begin{aligned}
\hat{B_1}^{\mu} &= [\frac{1}{(x_1-x_2)(x_1-x_3)}]\times \bar{R_B}(x_1) + \\
&\quad [\frac{1}{(x_2-x_1)(x_2-x_3)}]\times \bar{R_B}(x_2) + \\
&\quad [\frac{1}{(x_3-x_1)(x_3-x_2)}]\times \bar{R_B}(x_3) \bmod 7 \\
\hat{B_2}^{\mu} &= [\frac{(x_2+x_3)}{(x_1-x_2)(x_1-x_3)}]\times \bar{R_B}(x_1) + \\
&\quad [\frac{(x_1+x_3)}{(x_2-x_1)(x_2-x_3)}]\times \bar{R_B}(x_2) + \\
&\quad [\frac{(x_1+x_2)}{(x_3-x_1)(x_3-x_2)}]\times \bar{R_B}(x_3) \bmod 7 \\
\hat{B_3}^{\mu} &= [\frac{x_2x_3}{(x_1-x_2)(x_1-x_3)}]\times \bar{R_B}(x_1) + \\
&\quad [\frac{x_1x_3}{(x_2-x_1)(x_2-x_3)}]\times \bar{R_B}(x_2) + \\
&\quad [\frac{x_1x_2}{(x_3-x_1)(x_3-x_2)}]\times \bar{R_B}(x_3) \bmod 7
\end{aligned} \tag{7}
$$

$$
\hat{B}^{\mu} = \hat{B_1}^{\mu}\times 7^2 + \hat{B_2}^{\mu}\times 7 + \hat{B_3}^{\mu}. \tag{8}
$$

For example, the pixel values in the four-pixel block are represented as $\bar{P_1}^{\mu} = 163$, $\bar{P_2}^{\mu} = 150$, $\bar{P_3}^{\mu} = 171$, and $\bar{P_4}^{\mu} = 81$, respectively. Later on, we can get the values of $\bar{R_B}(x_1 = 2)$, $\bar{R_B}(x_2 = 3)$, and $\bar{R_B}(x_3 = 5)$ as 3, 6, and 3, respectively. Finally, with Equation (7), we can calculate the values of $\bar{B_1}^{\mu}$, $\bar{B_2}^{\mu}$, and $\bar{B_3}^{\mu}$ as 2, 0, and 2, respectively, and $\hat{B}^{\mu} = 100$. Then, we reshuffle the mean values back to be the original block location. Finally, we duplicate the mean value of block to expend to four pixels in the block to generate a new image WI.

If the block located at $(i, j)$-position is judged an inauthentic block, the four pixel values of the block are all replaced with the values of the pixels located at $(i, j)$-axis of WI'.

## 4 Experimental Results

In this section, we show our experimental results and the performances of our proposed scheme. Five grayscale images with size 256 × 256 pixels are as test images in the experiments, which are named "Barbara", "Baboon", "Boat", "Cartoon", and "Gold Hill". The visual quality measured by PSNR (peak-signal-to noise ratio) is used to evaluate the visual qualities between the original images and the watermarked images as listed in Table 1.

To measure the authentication and recovery abilities, five examples modified images are shown in Figures 3(a)-(e). After the authentication and reconstruction method, the located areas are shown in Figures 3(f)-(j), where the modified area is marked with black color. After recovery procedure, the recovery results are illustrated in Figures 3(k)-(o). The visual qualities of the recovery image are good to recognize what the original look like with naked eyes even the modified area is up to 50% (see Figure 3(l)).

## 5 Conclusion

In this paper, we proposed a self-authentication mechanism with recovery ability for digital images. With the proposed system, the image is authenticated whether the area is modified by comparing the generated authentication code and hidden authentication code together. Moreover, this scheme can reconstruct the secret image. In the experimental results, the proposed scheme shows the positive results to confirm its feasibility.

## Acknowledgments

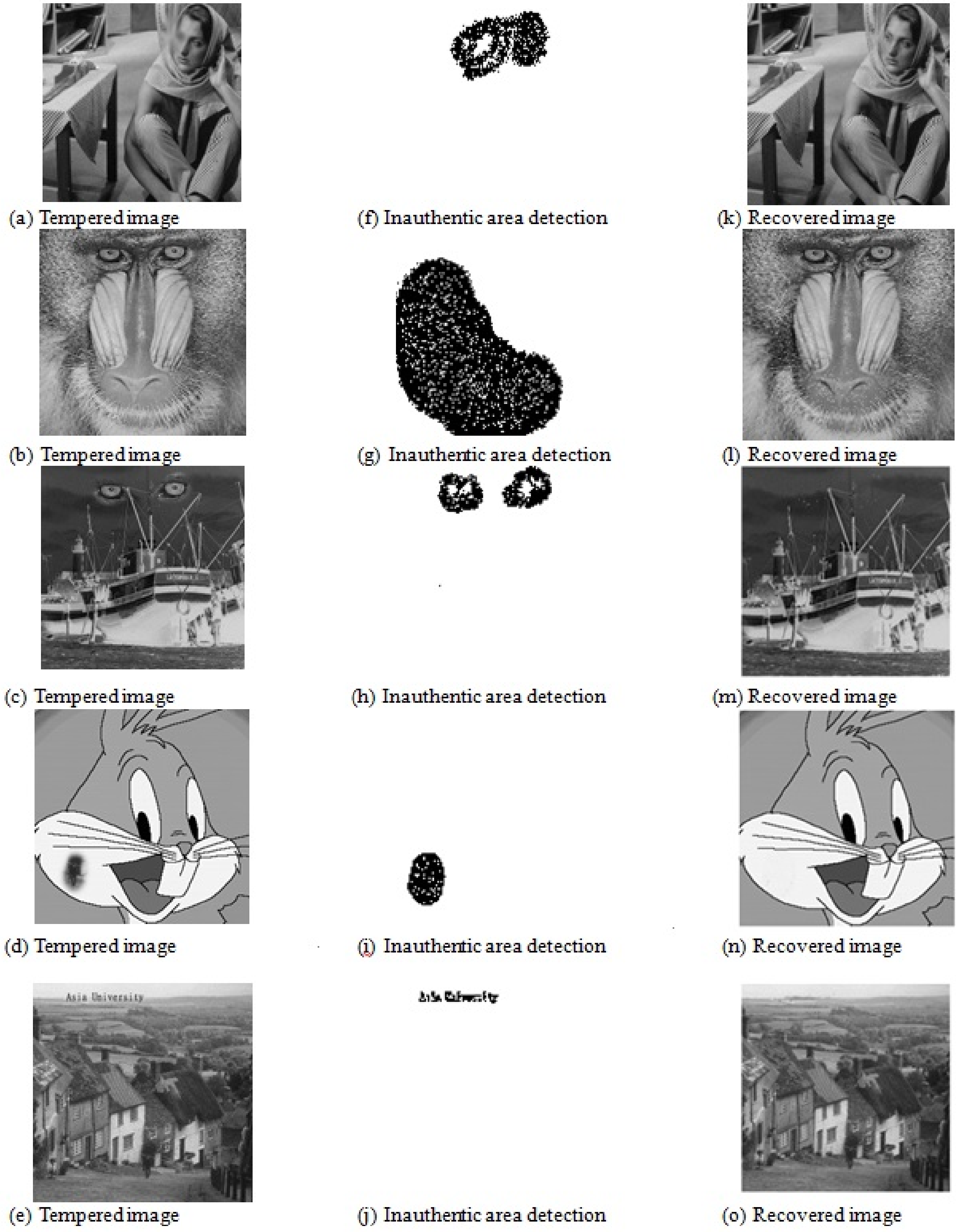
This work was supported by the Ministry of Science and Technology of Taiwan under Grant No. 103-2221-E-468-017.

## References

[1] C. C. Chang, Y. H. Hu, and T. C. Lu, "A watermarking-based image ownership and tampering authentication scheme," *Pattern Recognition Letters*, vol. 27, pp. 439-446, 2006.

[2] C. C. Chang, T. S. Nguyen, and C. C. Lin, "Reversible image hiding for high image quality based on histogram shifting and local complexity," *International Journal of Network Security*, vol. 16, no. 3, pp. 208-220, 2014.

[3] Y. H. Chen, P. Prangjarote, and C. Y. Lin, "Self-verifiable secret sharing scheme with locatability for halftone Images," *International Journal of Network Security*, vol. 17, no. 3, pp. 246-250, 2015.



(a) Tempered image (f) Inauthentic area detection (k) Recovered image

(b) Tempered image (g) Inauthentic area detection (l) Recovered image

(c) Tempered image (h) Inauthentic area detection (m) Recovered image

(d) Tempered image (i) Inauthentic area detection (n) Recovered image

(e) Tempered image (j) Inauthentic area detection (o) Recovered image

Figure 3: The tempered images and their corresponding recovery result

[4] H. He, J. Zhang, and H. M. Tai, "A wavelet-based fragile watermarking scheme for secure image authentication," in *Proceedings of the 5th International Workshop Dig. Watermarking*, vol. 4283, pp. 422-432, 2006.

[5] B. Karthikeyan, S. Ramakrishnan, V. Vaithiyanathan, S. Sruti, and M. Gomathymeenakshi, "An improved steganographic technique using LSB replacement on a scanned path image," *International Journal of Network Security*, vol. 16, no. 1, pp. 14-18, 2014.

[6] S. H. Liu, H. X. Yao, W. Gao, and Y. L. Liu, "An image fragile watermark scheme based on chaotic image pattern and pixel-pairs," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 869-882, 2007.

[7] H. Lu, R. Shen, and F. L. Chung, "Fragile watermarking scheme for image authentication," *Electronics Letters*, vol. 39, no. 12, pp. 898-900, 2003.

[8] P. D. Sheba Kezia Malarchelvi, "A semi-fragile image content authentication technique based on secure hash in frequency domain," *International Journal of Network Security*, vol. 15, no. 5, pp. 355-362, 2013.

[9] S. S. Sujatha and M. Mohamed Sathik, "A novel DWT based blind watermarking for image authentication," *International Journal of Network Security*, vol. 14, no. 4, pp. 223-228, 2012.

[10] S. Suthaharan, "Fragile image watermarking using a gradient image for improved localization and security," *Pattern Recognition Letters*, vol.25, pp. 1893-1903, 2004.

[11] D. Wang, C. C. Chang, Y. Liu, G. Song, Y. Liu, "Digital image scrambling algorithm based on Chaotic sequence and decomposition and recombination of pixel values," *International Journal of Network Security*, vol. 17, no. 3, pp. 322-327, 2015.

[12] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image autheticataion and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1593-1601, 2001.

[13] X. Zhang and S. Wang, "Fragile watermarking scheme using a hierarchical mechanism," *Signal Processing*, vol. 89, no. 4, pp. 675-679, 2009.

**Yi-Hui Chen** received B.S. and M.S. degrees in information management from the Chaoyang University of Technology in 2001 and 2004, respectively. In 2009, she earned her Ph.D. degree in computer science and information engineering at the National Chung Cheng University. From 2009 to 2010, she worked with Academia Sinica as a post-doctoral fellow. Later, she worked at IBM's Taiwan Collaboratory Research Center as a Research Scientist. She is now an assistant professor with the Department of Applied Informatics and Multimedia, Asia University. Her research interests include image processing, watermarking, steganography, and XML techniques.

**Chih-Yang Lin** received the B.S. degree in computer science and information engineering from Tung-Hai University, Taichung, in 1998, the master degree in information management from National Chi-Nan University, Nantou, in 2000. In 2006, he received a Ph.D. degree from Dept. Computer Science and Information Engineering at National Chung-Cheng University, Chiayi. After graduated, he servered in Advanced Technology Center of Industrial Technology Research Institute of Taiwan (ITRI) from 2007 to 2009. Then, he joined the Institute of Information Science (IIS), Academia Sinica, as a postdoctoral fellow. Currently, he is an Assistant Professor in the Department of Computer Science and Information Engineering, Asia University. His research interests include computer vision, digital rights management, image processing, and data mining.

**Wanutchaporn Sirakriengkrai** received B.S. degree in Humanities from Bangkok University in 2010. Now, she is a master degree student in computer science at Asia University. Her research focus on image processing and image authentication.

**I-Chun Weng** studied Computer Science and Information Engineering at Asia University. Her research interests focus on image processing and computer vision.

# A Study of Relationship between RSA Public Key Cryptosystem and Goldbach's Conjecture Properties

Chenglian Liu[1]*, Chin-Chen Chang[2,3], Zhi-Pan Wu[1], Shi-Lin Ye[1]
*(Corresponding author: C. C. Chang)*

Department of Computer Science, Huizhou University[1]
Huizhou 516007, China.
Department of Information Engineering and Computer Science, Feng Chia University[2]
Taichung 40724, Taiwan.
The Department Computer Science and Information Engineering, Asia University[3]
Taichung 41354, Taiwan.
(Email: alan3c@gmail.com)

*(Received Dec. 25, 2014; revised and accepted Mar. 4 & Apr. 21, 2015)*

## Abstract

The Goldbach's conjecture has plagued mathematicians for over two hundred and seventy years. Whether professionals or amateur enthusiasts, all have been fascinated by this question. Why do mathematicians have no way to solve this problem? Till now, Chen has been recognized for the most concise proof his "1 + 2" theorem in 1973. In this article the authors will use elementary concepts to describe and indirectly prove the Goldbach conjecture.

*Keywords: AKS algorithm, number axis, symmetrical primes*

## 1 Introduction

Until now, the best proof of the theorem is by Chen [3] in 1973 that states every large even integer can be written as the sum of a prime and the product of at most two primes. Recently, Bournas [2] proposed his contribution that proves the conjecture is true for all even integers greater than 362. Silva et al. [6] describes how the even Goldbach conjecture was confirmed to be true for all even numbers not larger than $4 \cdot 10^{18}$ and the odd Goldbach conjecture is true up to $8.37 \cdot 10^{26}$. Lu [16] showed an even integer $x$ at most $\mathcal{O}(x^{0.879})$ can not be written as a sum of two primes. On the other hand, Zhang [26] proved that there are infinitely many pairs of primes that differ by less than $7 \cdot 10^7$. Zhang's result is a huge step forward in the direction of the twin prime conjecture. Some people in related research also gave good contributions [8–11, 13, 18, 22, 25].

In this paper, the authors will introduce the fundamental concepts rather than the entire proof in its complexity.

## 2 Review of Goldbach conjecture issue

The (strong) Goldbach conjecture states that every even integer $N$ greater than six can be written as the sum of two primes such as

$$\begin{aligned} 138 &= 131+7 \\ &= 127+11 \\ &= 109+29 \\ &= 107+31 \\ &= 101+37 \\ &= 97+41 \\ &= 79+59 \\ &= 71+67. \end{aligned}$$

The expression of a given even number as a sum of two primes is called a 'Goldbach partition' of that number. For example: The integer 138 can be expressed in 8 ways. We say the GC number can be described in the form as

$$GC = P_i + P_j \longmapsto (P_i - 2n) + (P_j + 2n), \tag{1}$$

where $P_i$ and $P_j$ are both primes. Let $R(n)$ be the number of representations of the Goldbach partition where $\prod_2$ is the twin prime constant [14], say $R(n) \sim$

*Part of this work was carried out while the first author was at Longyang University supported by the information security team project. Email:chenglian.liu@gmail.com

$2\prod_2 \left(\prod_{P_k|n,k=2}\right) \frac{P_k-1}{P_k-2} \int_2^n \frac{dx}{(\ln x)^2}$. Ye and Liu [24] also gave the estimation formula $G(x) = 2C\prod_{p\geq 3}\frac{(p-1)}{(p-2)} \cdot \frac{(Li(x))^2}{x} + \mathcal{O}(x \cdot e^{-c\sqrt{\ln x}})$.

## 2.1 The RSA Cryptosystem

The RSA algorithm [21] is well known public key cryptosystem. It is widely used many application such as traitor tracing scheme [23], multi-secrect sharing scheme [5], and anonymous multi-receive encryption scheme [12] so on. We briefly introduce the principle of RSA in this subsection. The signer prepares the prerequisite of an RSA signature: two distinct large primes $p$ and $q$, $n = pq$, Let $e$ be a public key so that $\gcd(e, \phi(n)) = 1$, where $\phi(n) = (p-1)(q-1)$, then calculate the private key $d$ such that $ed \equiv 1 \pmod{\phi(n)}$. The signer publishes $(e, n)$ and keeps $(p, q, d)$ secret. The notations are the same as in [21].

**RSA Encryption and Decryption:**
In RSA public-key encryption, Alice encrypts a plaintext $M$ for Bob using Bob's public key $(n, e)$ by computing the ciphertext

$$\begin{aligned} C &\equiv M^e \pmod{n}, \\ M &\equiv C^d \pmod{n}, \end{aligned}$$

where $n$, the modulus, is the product of two or more large primes, and $e$, the public exponent, is an (odd) integer $e \geq 3$ that is relatively prime to $\phi(n)$, the order of the multiplicative group $\mathbb{Z}_n^*$. The signer uses private key $d$ to decrypt message $M$ from the ciphertext $C$.

**RSA Digital Signature:**

$$s \equiv M^d \pmod{n},$$

where $(n, d)$ is the signer's RSA private key. The signature is verified by recovering the message $M$ with the signer's RSA public key $(n, e)$:

$$M \equiv s^e \pmod{n}.$$

## 2.2 The Relationship of the Goldbach's Conjecture and the RSA Cryptosystem

Constant [4] proposed the algebra factoring of the cryptography modulus and proof of Goldbach's conjecture. He connected each relationship. His methodology is described as follows:
Since we know the modulus $n = p \cdot q$, we assume

$$s = p + q.$$

Step 1. Compute

$$p^2 - sp + n = 0.$$

Step 2. Compute

$$p, q = \frac{1}{2}(s \pm c) \qquad (2)$$

since

$$c = \sqrt{s^2 - 4n}. \qquad (3)$$

Step 3. Compute $s^2 = c^2 + 4n$, or we can reexpress as

$$c^2 = s^2 - 4n.$$

**Example 1:**
We assume $n = 721801$, then $4n = 4 \cdot 721801 = 2887204$. We also compute $\sqrt{4n} \approx 1699.177$ since $s^2 > 4n$, we therefore start the integer $s$ by 1700. From Equation (2) and Equation (3), we have $s = 1802$, and $c = 600$, to calculate the following table.

Table 1: $n = 721801$

| Times | $s$ | $s^2$ | $4n$ | $c^2$ | $c$ |
|---|---|---|---|---|---|
| 1 | 1700 | 2890000 | 2887204 | $\sqrt{2796}$ | 52.87 |
| 2 | 1702 | 2896804 | 2887204 | $\sqrt{9600}$ | 97.97 |
| 3 | 1704 | 2903616 | 2887204 | $\sqrt{16412}$ | 128.10 |
| 4 | 1706 | 2910436 | 2887204 | $\sqrt{23232}$ | 152.42 |
| 5 | 1708 | 2917264 | 2887204 | $\sqrt{300600}$ | 173.37 |
| 6 | 1710 | 2924100 | 2887204 | $\sqrt{36896}$ | 192.08 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 51 | 1800 | 3240000 | 2887204 | $\sqrt{352796}$ | 593.96 |
| 52 | 1802 | 3247204 | 2887204 | $\sqrt{360000}$ | 600 |

$$\begin{aligned} p &= \frac{1802 + 600}{2} = 1201, \\ q &= \frac{1802 - 600}{2} = 601. \end{aligned}$$

We obtain $p = 1201$, and $q = 601$. The result is as shown in Table 1.

**Example 2:**
We assume $n = 321907$ where $s^2 > 4n$, namely $4n = 4 \cdot 321907 = 1287628$. Since $\sqrt{1287628} \approx 1134.73$, we therefore start the integer $s$ by 1136. From above it is stated, $c$ must be an integer. Hence, we assume $s = 1148$ and set $c = 174$. From Equation (2) and Equation (3), we have

$$\begin{aligned} p &= \frac{1148 + 174}{2} = 661, \\ q &= \frac{1148 - 174}{2} = 487. \end{aligned}$$

We obtain $p = 661$, and $q = 487$. The result is as shown in Table 2. When the modulus $n$ goes up to 1024-bits or greater than 2048-bits length, is this methodology still efficient? This is an interesting question.

Table 2: $n = 321907$

| Times | $s$ | $s^2$ | $4n$ | $c^2$ | $c$ |
|---|---|---|---|---|---|
| 1 | 1136 | 1290496 | 1287628 | $\sqrt{2868}$ | 53.55 |
| 2 | 1138 | 1295044 | 1287628 | $\sqrt{7416}$ | 86.11 |
| 3 | 1140 | 1299600 | 1287628 | $\sqrt{11972}$ | 109.41 |
| 4 | 1142 | 1304164 | 1287628 | $\sqrt{16563}$ | 128.59 |
| 5 | 1144 | 1308736 | 1287628 | $\sqrt{21108}$ | 145.28 |
| 6 | 1146 | 1313316 | 1287628 | $\sqrt{25688}$ | 160.27 |
| 7 | 1148 | 1317904 | 1287628 | $\sqrt{30276}$ | 174 |

## 3 Our Analysis

In this section, we introduce another methodology that analyzes the Goldbach's conjecture properties and the relationship with twin prime.

### 3.1 The Goldbach's Conjecture Properties

In this subsection, the authors describe the Goldbach's conjecture properties. Notations are described in the following.

**Notations:**

$P_n$: The $n$th prime number.

$g_p$: Smallest prime factor of number $m$.

$P[m]$: Largest prime factor of $m$.

$P_0[m]$: Smallest prime factor of $m > 1$.

$d_k$: $= P_j - P_i$, gap or distance between two primes, it should be an even integer.

$\pi(x)$: The number of primes $p$, $p \leq x$.

$G(x)$: The number of Goldbach partition.

$GC$: An even number for the Goldbach Conjecture (GC) number.

$PG$: An integer for the prime gaps (PG) number.

$M$: Denotes $M = \frac{GC}{2}$.

$\overline{P_iM}$: A distance value from point $P_i$ to point $M$, this value differs from $d_k$ if $M$ is not a prime.

$\overline{MP_j}$: A distance value from point $M$ to point $P_j$, this value differs from $d_k$ if $M$ is not a prime.

$SPN$: Assume $P_i$ and $P_h$ are prime number pairs. $M$ is the midpoint between $P_i$ and $P_h$, where $M$, $P_i$, $P_h$ lie on the X axis; say $P_i$ and $P_h$ are symmetric prime numbers to integer $M$ on the X axis.

$2n|\overline{P_iM}$: The $2n$ divide the $\overline{P_iM}$.

Some basic properties are shown as follows:

**Property 1.** odd + even = odd.

**Property 2.** even + even = even.

**Property 3.** odd + odd = even.

**Property 4.** even − even = even.

**Property 5.** odd − odd = even.

**Property 6.** even − odd = odd.

**Property 7.** even · even = even.

**Property 8.** odd · even = even.

**Property 9.** odd · odd = odd.

The relationship diagram is shown in Figure 1.

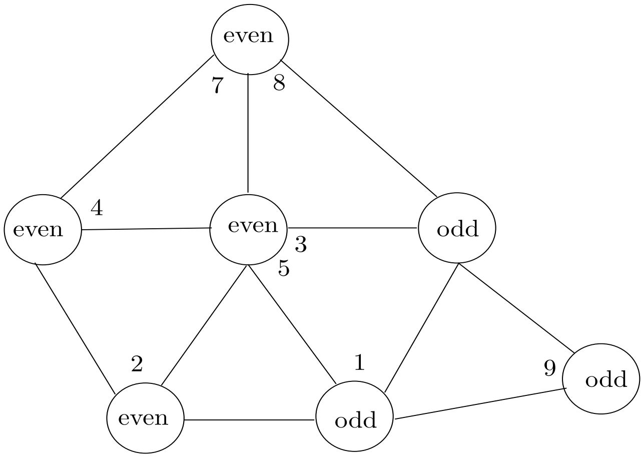


Figure 1: The odd and even numbers relationship of properties in arithmetic

In this article, we classify the Goldbach Conjecture (GC) into three categories. The fundamental concepts in detail are shown in Figure 2. For convenience, we used the notation Case 1, Case 2 and Case 3 to describe the following scenarios. We suppose an integer $GC$, where $GC \geq 6$ and it is an even positive number, there also exists an integer $M$, where $M = \frac{GC}{2}$. We use an X-axis line to express distance, see Figure 3.

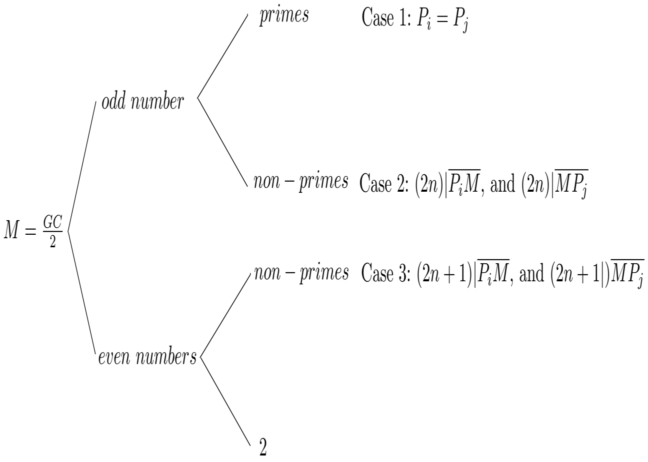

$M = \frac{GC}{2}$
odd number
primes
Case 1: $P_i = P_j$
non − primes
Case 2: $(2n)|\overline{P_iM}$, and $(2n)|\overline{MP_j}$
even numbers
non − primes
Case 3: $(2n+1)|\overline{P_iM}$, and $(2n+1|)\overline{MP_j}$
2

Figure 2: The Goldbach conjecture's situation case

Case 1: If $M$ is a prime, then there exists a prime number, say $P_i$ where $P_i = P_j$ and located on $M$ point at $X$ axis (See Figure 4).

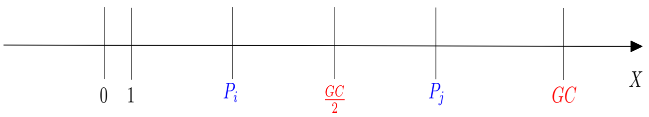


Figure 3: The $X$-axis of number line

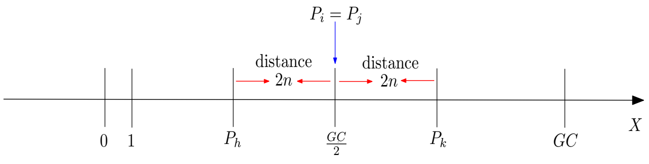


Figure 4: Case 1 situation

Case 2: If $M$ is not a prime, and is an odd number, there exists at least one pair of symmetrical primes. Say $P_i$ and $P_j$, where the distance is $\overline{P_iM} = \overline{MP_j}$, and $2n|\overline{P_iM}$, $2n|\overline{MP_j}$ (See Figure 5).

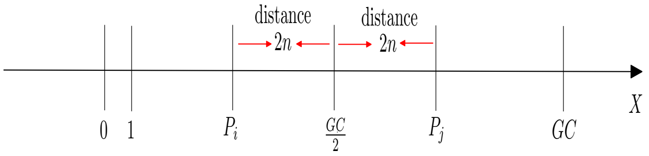


Figure 5: Case 2 situation

Case 3: If $M$ is not a prime, and is an even number, there exists at least one pair of symmetrical primes. Say $P_i$ and $P_j$ where the distance is $\overline{P_iM} = \overline{MP_j}$, and $2n+1|\overline{P_iM}$, $2n+1|\overline{MP_j}$ (See Figure 6).

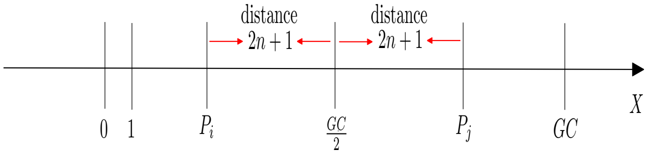


Figure 6: Case 3 situation

**Theorem 1** (Bertrand-Chebyshev Theorem). *For any real number $n$, where $n \geq 1$, there always exists at least a prime between the interval $n$ and $2n$.*

*Proof.* We suppose that

$$\begin{aligned}\binom{2n}{n} &\leq \prod_{p\leq\sqrt{2n}} P^r \prod_{\sqrt{2n}<p\leq\frac{3}{2}n} P \prod_{m<p\leq 2n} P\\ &\leq \prod_{p\leq\sqrt{n}} (2n) \prod_{\sqrt{2n}<p\leq m} P \prod_{m<p<2m} P. \qquad (4)\end{aligned}$$

For each $n$, where $1 \leq n < 4010$, such as 2, 3, 5, 7, 13, 23, 43, 83, 163, 317, 631, 1259, 2503, …, 3967, 3989, 4001, 4003, 4007. We choose a small prime $p$, and another greater than $n$ say $p'$. The relationship is as follows:

$$p \leq n \leq p' \leq 2p \leq 2n. \qquad (5)$$

Thus, this finishes the proof. □

**Proposition 1.** *If $M = \frac{GC}{2}$, where $M$ is a prime, say $M = P_i = P_j$, and $P_i$ located on $M$ point at $X$ axis. There exists at least one pair of symmetrical primes $P_h$ and $P_k$, where the distance value $\overline{P_hM} = \overline{MP_k}$.*

*Proof.* We assume $M$ is prime, then $M - P_h = \overline{P_hM}$ is also an even integer, according to Property 5. The odd integers are subtracted to give an even integer. There are two symmetrical prime numbers, say $P_h$ and $P_k$ located on the two sides of $M$ at the center point position. The distance $\overline{P_hM}$ is equal to distance $\overline{MP_k}$, divided by $2n$. If $\frac{P_h+P_k}{2} = M$ while $P_h \neq P_i \neq P_k$, it also matches $P_h + P_k = GC$. Thus, we have obtained the first solution $M = P_i = P_j$ if and only if $M$ is a prime. The second solution is $P_h + P_k = GC$ if and only if $P_h$ and $P_k$ are both primes. □

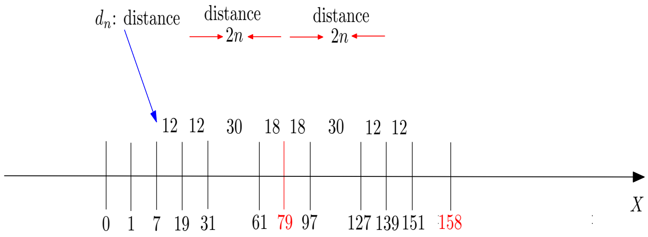


Figure 7: An example of Case 1 situation

Suppose $GC = 158$, and $\frac{GC}{2} = 79$.

$$\begin{aligned}158 &= 7 + 151\\ &= 19 + 139\\ &= 31 + 127\\ &= 61 + 97\\ &= 79 + 79.\end{aligned}$$

**Proposition 2.** *If $M$ is not a prime, but is an odd number, there exists at least two prime numbers, say $P_h$ and $P_k$ that are located on either side of the center point $M$. The distance from $P_i$ to $M$ is equivalent to that from $M$ to $P_j$.*

*Proof.* We assume $M$ is an odd number, then $M - P_i = P_j - M$. As stated previously $P_i + P_j = 2M = GC$, but $P_i \neq P_j$. From Property 5, the odd integers are subtracted to give an even integer. Thus, we have the value $\overline{P_iM}$ of distance from $P_i$ to $M$ must be an even integer, and is divided $2n$. On the other hand, there is a similar situation from $M$ to $P_j$ since $2n|\overline{P_iM}, 2n|\overline{MP_j}$ while $P_i \neq P_j$. We have $P_i + P_j = 2M = GC$, because $P_i \neq P_j$ and $P_i < M < P_j$. This is one solution of symmetrical primes. Case 1 is a special situation of Case 2. □

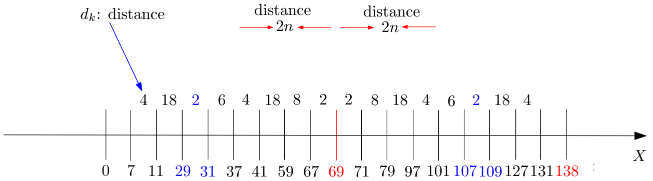


Figure 8: An example of Case 2 situation

Suppose $GC = 138$, and $\frac{GC}{2} = 69$.

$$\begin{aligned} 138 &= 131 + 7 \\ &= 127 + 11 \\ &= 109 + 29 \\ &= 107 + 31 \\ &= 101 + 37 \\ &= 97 + 41 \\ &= 79 + 59 \\ &= 71 + 67. \end{aligned}$$

**Proposition 3.** *If $M = \frac{GC}{2}$, is not a prime, but is an even number, there exists at least two primes, say $P_i$ and $P_j$ located on either side of $M$ centerpoint position, where the distance $\overline{P_iM}$ equals $\overline{MP_j}$, $2n+1|\overline{P_iM}$, $2n+1|\overline{MP_j}$.*

*Proof.* We assume $M$ is not a prime and is an even number. According to Property 6, the even number is subtracted from the odd number and the result is an odd number. We, therefore, know this distance value must be an odd integer while $P_j \neq P_j$. Hence, the relationship as $P_i < M < P_j$. Since $\overline{P_iM} = \overline{MP_j}$. We have $P_i + P_j = 2M = GC$; however, $P_i \neq P_j$. Thus, we obtained one solution where two primes are symmetrical about the point of $M$ on the X axis line. If and only if $n = 0$, where $M - P_i$ equals $P_j - M$, it has $P_j - P_i = 2$ since $P_i + P_j = 2M = GC$, say $(P_i,\ P_j)$ are twin primes. The twin prime is also a special situation of Case 3. □

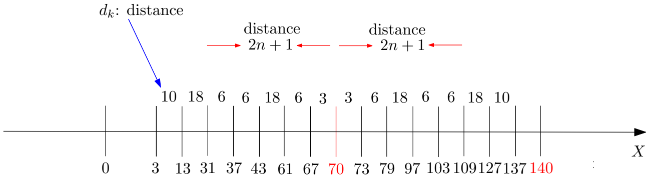


Figure 9: An example of Case 3 situation

Suppose $GC = 140$, and $\frac{GC}{2} = 70$.

$$\begin{aligned} 140 &= 3 + 137 \\ &= 13 + 127 \\ &= 31 + 109 \\ &= 37 + 103 \\ &= 43 + 97 \\ &= 61 + 79 \\ &= 67 + 73. \end{aligned}$$

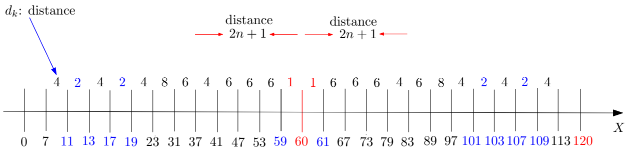


Figure 10: An example of twin prime situation

Suppose $GC = 120$, and $\frac{GC}{2} = 60$.

$$\begin{aligned} 120 &= 7 + 113 \\ &= 11 + 109 \\ &= 13 + 107 \\ &= 17 + 103 \\ &= 19 + 101 \\ &= 23 + 97 \\ &= 31 + 89 \\ &= 37 + 83 \\ &= 41 + 79 \\ &= 47 + 73 \\ &= 53 + 67 \\ &= 59 + 61. \end{aligned}$$

**Theorem 2.** *For all prime numbers that are greater than 3, the prime gap (PG, or distance) is an even integer.*

*Proof.* For any prime numbers that are greater than 3, the PG should be an odd number. From Property 5, the answer is an even number when two odd numbers are subtracted from each other. The prime gap is an even number if the prime is greater than 3. Suppose two odd numbers $p$ and $q$, where $p < q$, and $p \neq q$. Since

$$\begin{aligned} p &\equiv 1 \pmod 2 \\ \textit{and } q &\equiv 1 \pmod 2, \end{aligned}$$

we obtained $|p - q| \equiv 0 \pmod 2$. □

**Lemma 1.** *We suppose the prime gap PG is a positive integer. From Theorem 2, the $\frac{PG}{2}$ has two results, it may have an even number, or may have an odd number. We rewrite the expression as*

$$\frac{PG}{2} \begin{cases} \equiv 0 \pmod 2, \textit{ this is an even number.} \\ \equiv 1 \pmod 2, \textit{ this is an odd number.} \end{cases}$$

*When $\frac{PG}{2} \equiv 0 \pmod 2$, is an even integer; and $\frac{PG}{2} \equiv 1 \pmod 2$ is an odd integer.*
*Let $d = \frac{PG}{2}$, it then*

$$q - d = \begin{cases} \textit{even number.} \\ \textit{odd number.} \end{cases}$$

*We assume $d = \frac{PG}{2}$, and $q - d = s$.*

1) *If $d$ is an odd integer, from Property 5, the $s$ should be an even integer.*

*2) If $d$ is an even integer, from Property 6, the $s$ should be an odd integer.*

**Theorem 3** (Symmetric Prime Number Theorem)**.** *For any two prime numbers $p$ and $q$, $p < q$ that are greater than 3, with the $X$ axis as the line of symmetry, the two prime numbers should be located on both sides of an integer $M$, the distance from $p$ to $M$ and $M$ to $q$ are proportionally equal.*

*Proof.* As known,

$$(q - M) = (M - p),$$

since

$$(q + p) = 2M.$$

From Theorem 1, there exists at least a prime between $M$ and $2M$. In other words, there also exists at least a prime between $\frac{M}{2}$ and $M$. Hence, there are two prime numbers located on the X axis line between $\frac{M}{2}$ and $2M$. It can be seen, the primes $p$ and $q$ are symmetrical to $M$. If not, the $(q - p) = (M - p)$ is a contradiction. □

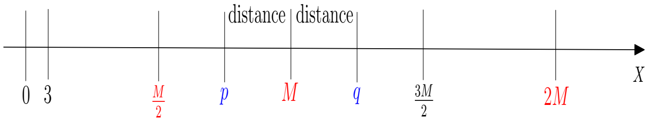


Figure 11: The symmetric primes on the X axis situation

There is some related literature about prime symmetric problems in [7,17,19,20], but slightly different than what is discussed in this article.

## 3.2 The Goldbach's Conjecture and the Twin Prime Relationship

In this subsection, the authors describe a relationship of Goldbach's conjecture and twin prime. Previously, we listed an example of a special situation in Case 3, and drew a diagram in Figure 10. Here, we discuss in depth this issue. We describe the conception of prime combinations in Goldbach's conjecture. From Equation (1), rewrite as the following:

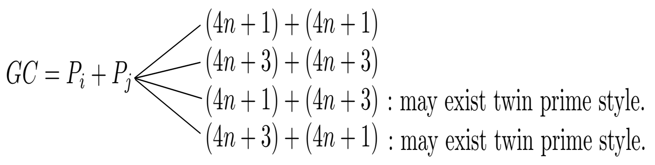

$GC = P_i + P_j$
$(4n+1)+(4n+1)$
$(4n+3)+(4n+3)$
$(4n+1)+(4n+3)$ : may exist twin prime style.
$(4n+3)+(4n+1)$ : may exist twin prime style.

Figure 12: The twin prime of Goldbach's conjecture on the X axis situation

$$P_i + P_j = \begin{cases} (4n+1)+(4n+1), \text{ are both '+1' form.} \\ (4n+3)+(4n+3), \text{ are both '+3' form.} \\ (4n+1)+(4n+3), \text{ mixed '+1' and '+3' form.} \end{cases}$$

**Theorem 4.** *For each twin prime pair $(P_i, P_j)$ where the integers are greater than or equal to $(5, 7)$, say $(P_i, P_j) \geq (5, 7)$. There must belong this type of '$(4n+1)+(4n+3)$' or '$(4n+3)+(4n+1)$' forms.*

*Proof.* For each twin prime pair $(P_i, P_j)$ where the values are greater than or equal to $(5, 7)$. We assume an integer $n$, where $n \geq 1$, namely

$$(4n+1) - (4n+1) = 0 \pmod 4,$$

and

$$(4n+3) - (4n+3) = 0 \pmod 4.$$

On the other hand,

$$(4n+3) - (4n+1) = 2 \pmod 4,$$

or

$$(4n+1) - (4n+3) = |-2| \equiv 2 \pmod 4.$$

This is to say, the twin prime pair $(P_i, P_j)$ must be expressed as the form of '$(4n+1)+(4n+3)$' or '$(4n+3)+(4n+1)$'. Otherwise, it is a contradiction. □

The relationship of twin prime pair $(P_i, P_j)$, as shown in Figure 13 and Figure 14.

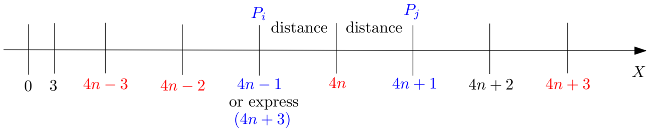


Figure 13: An relationship of twin prime situation I

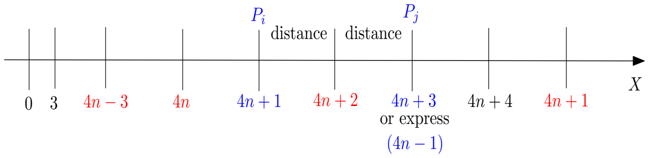


Figure 14: An relationship of twin prime situation II

**Proposition 4.** *If $P_i + P_j \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 4 \pmod 8$, and $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 2 \pmod 8$ or $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 6 \pmod 8$, there may exist a twin prime where the $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ is $(4n+1)+(4n+3)$ form.*

*Proof.* As known from Proposition 3, $\frac{P_i+P_j}{2}$ is an even number. Otherwise, it is a contradiction. According to Property 6:

$$\begin{cases} \frac{P_i+P_j}{2} - 1 \text{ is an odd number.} \\ \frac{P_i+P_j}{2} + 1 \text{ is an odd number too.} \end{cases}$$

Note that $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 6 \pmod 8$, we see the $\frac{P_i+P_j}{2}$ is $4n+2$ form. Therefore, the $\frac{P_i+P_j}{2} - 1$

is $4n+1$ form, and $\frac{P_i+P_j}{2}+1$ is $4n+3$ form.
Since $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 2 \pmod 8$, by Theorem 4, we know $(\frac{P_i+P_j}{2}-1, \frac{P_i+P_j}{2}+1)$ is $(4n+1)+(4n+3)$ form. □

**Proposition 5.** *If $P_i+P_j \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$, and $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$ or $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 4 \pmod 8$, there may exist a twin prime where $(\frac{P_i+P_j}{2}-1, \frac{P_i+P_j}{2}+1)$ is $(4n+3)+(4n+1)$ form.*

*Proof.* As known, the $\frac{P_i+P_j}{2}$ is an even number. Since $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$. We see the $\frac{P_i+P_j}{2}$ is $4n$ form. Hence $\frac{P_i+P_j}{2}-1$ is $4n+3$ form. Therefore $\frac{P_i+P_j}{2}+1$ is $4n+1$ form. Now, as $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$, the $\frac{P_i+P_j}{2}$ is $4n$ form too. Thus, the $\frac{P_i+P_j}{2}+1$ is $4n+1$ form. This inference is consistent with the above statement. □

**Proposition 6.** *If $\frac{P_i+P_j}{2}$ is prime, the $P_i+P_j$ can not be combined with $(4n+1)+(4n+3)$ or $(4n+3)+(4n+1)$ forms. It can be represented as $(4n+1)+(4n+1)$ or $(4n+3)+(4n+3)$ forms. It is impossible to have $(4n+3)+(4n+1)$ or $(4n+1)+(4n+3)$ forms.*

*Proof.* We suppose $P_i, P_h$ and $P_j$ are three primes, where $P_h = \frac{P_i+P_j}{2}$.
By Lemma 1, there exists an integer $s$, where $s = P_h - P_i$. Since $P_j = P_h + s$ and $2P_h = P_i + P_j$, if $P_h$ is $4n+1$ form, then this is $(4n+1)+(4n+1)$ form, say $P_h + P_j$. From Proposition 1, if and only if $P_h$ is $4n+1$ form, then $P_h - s = P_i$, where $s$ is an even number. We rewrite it as follows:
$(4n+1)-2n = P_i$ is $4n+1$ form (while $n=0$).
Alternatively, $(4n+1)+2n = P_j$ is $4n+3$ form (while $n=1$).
If and only if $P_h$ is $4n+3$ form, then $P_h + s = P_j$. We rewrite the expression as below: $(4n+3)+2n = P_j$ is $4n+3$ form (while $n=0$).
On other side, $(4n+3)-2n = P_j$ is $4n+1$ form (while $n=1$). □

In summary, Goldbach's conjecture $\supseteq (4n+1)+(4n+3)$ $\subset$ twin prime.

### 3.3 The Relationship between $G(x)$ and $\pi(x)$ in Goldbach's Conjecture

In Table 3, the $G(x)$ is the number of prime pairs. For example, the positive integer 25,300 has 314 prime pairs matched with the Goldbach's rule. And the integer 253,000 has 2011 prime pairs matches. When the integer is approaching infinity, the $G(x)$ is also increased. However, Items 5, 9, 11 and 14 are exceptions. Note that a pattern begins to surface beginning with the 4th item. The $G(x)$ term value is between 5 and 6 for every two rows following. When the positive integer is approaching infinity, then the number of prime numbers $\pi(x)$ also increasing; it shows a very steady positive growth. However the $G(x)$ does not follow this rule. Different even numbers $GC$ for different swayed Goldbach partitions. There is no any strong relevance between each number $GC_i$ to the other number $GC_j$. Hence, there are no rules to predict this status. The experimental results are shown in Table 3 and Figure 15.

Table 3: The relationship of Goldbach partition $G(x)$ with $\pi(x)$

| item | Positive Integer | $G(x)$ | $\pi(x)$ | $\frac{\pi(x)}{G(x)}$ |
|---|---|---|---|---|
| 1 | 12650 | 186 | 1510 | 8.11 |
| 2 | 25300 | 314 | 2787 | 8.87 |
| 3 | 50600 | 553 | 5190 | 9.38 |
| 4 | 75900 | 1478 | 7473 | 5.05 |
| 5 | 101200 | 918 | 9691 | 10.55 |
| 6 | 126500 | 1140 | 11864 | 10.40 |
| 7 | 151800 | 2635 | 14007 | 5.31 |
| 8 | 177100 | 1802 | 16091 | 9.92 |
| 9 | 202400 | 1669 | 18178 | 10.89 |
| 10 | 227700 | 3688 | 20243 | 5.48 |
| 11 | 253000 | 2011 | 22280 | 11.07 |
| 12 | 278300 | 2130 | 24301 | 11.40 |
| 13 | 303600 | 4676 | 26289 | 5.62 |
| 14 | 318950 | 2059 | 27520 | 13.36 |
| 15 | 331600 | 2160 | 28533 | 13.20 |
| 16 | 344250 | 4652 | 29521 | 6.34 |
| 17 | 356900 | 2356 | 30512 | 12.95 |
| 18 | 369500 | 2321 | 31488 | 13.56 |
| 19 | 382200 | 6325 | 32460 | 5.13 |
| 20 | 394850 | ⋮ | ⋮ | ⋮ |
| 21 | 407500 | ⋮ | ⋮ | ⋮ |
| 22 | 420150 | 5264 | 35398 | 6.72 |

Note: this table does not include the prime number 2

**Open problems:**

1) How did we know the $\frac{GC}{2}$ is a prime number? The AKS algorithm [1] determines whether a number is prime or composite within polynomial time, it may be a discrepancy in the method. Lenstra and Pomerance [15] primality testing is other solution.

2) If the twin prime problem is solved, could it also solve the Goldbach's conjecture? The authors doubts this is the case. The twin prime situation is just a special case in Goldbach's conjecture.

3) If the puzzle of prime numbers is solved, will it may also solve the number of Goldbach partition?

## 4 Conclusions

We clearly described two examples of relationship between RSA and Goldbach conjecture; this method suc-

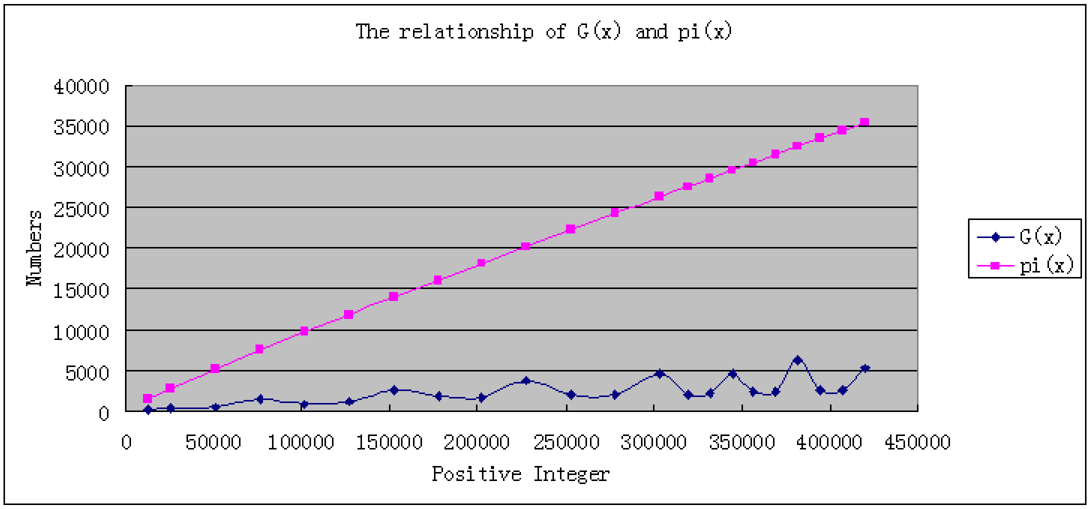


Figure 15: A relationship of $G(x)$ with $\pi(x)$ in positive integers

cessfully attack the RSA cryptosystem. Our contribution are useful to understand other algebra factoring methodology, when the modulus $n$ goes up to over 1024 bits length, does it still efficiency to factor? It becomes to our future work. On the other hand, for the prime number gaps problem, Zhang has a very good result. However, it is still far from a way to solve the Goldbach conjecture. The authors pointed out the prime symmetrical situation, may be useful to assist in understanding about the Goldbach conjecture, even though they did not offer a general formula on the Goldbach partition. The prime symmetrical property may also solve the puzzle of prime numbers.

## Acknowledgments

The authors would like to thank the reviewers for their comments that help improve the manuscript. The authors would like to thank the reviewers for their comments that help improve the manuscript. This work is partially supported by the project from department of education of Fujian province under the number JA12351 with JK2013062 and also partially supported from Guandong province special funds to foster the students' science and technology innovation under the number 139785 and 139802 (Climbing program funds).

## References

[1] M. Agrawal, N. Kayal, and N. Saxena, "Primes is in P," *Annals of Mathematics*, vol. 160, pp. 781–793, 2004.

[2] R. M. Bournas, "The strong goldbach conjecture: Proof for all even integers greater than 362," Sep. 2013. (http://arxiv.org/vc/arxiv/papers/1303/1303.4649v1.pdf)

[3] J. R. Chen, "On the representation of a larger even integer as the sum of a prime and the product of at more two primes," *Sciencia Sinica*, vol. 16, pp. 157–176, 1973.

[4] J. Constant, "Algebraic factoring of the cryptography modulus and proof of Goldbach's conjecture," July 2014. (http://www.coolissues.com/mathematics/Goldbach/goldbach.htm)

[5] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Network Security*, vol. 2, no. 2, pp. 69–72, 2015.

[6] T. O. e Silva, S. Herzog, and S. Pardi, "Empirical verification of the even Goldbach conjecture and computation of prime gaps up to $4 \cdot 10^{18}$," *Mathematics of Computation*, vol. 83, pp. 2033-2060, 2014.

[7] P. Fletcher, W. Lindgren, and C. Pomerance, "Symmetric and asymmetric primes," *Journal of Number Theory*, vol. 58, pp. 89–99, 1996.

[8] J. Ghanouchi, "A proof of Goldbach and de Polignac conjectures," 2015. (http://unsolvedproblems.org/S20.pdf)

[9] D. A. Goldston, J. Pintz, and C. Y. Yildirim, "Primes in tuples I," *Annals of Mathematics*, vol. 170, pp. 819–862, Sep. 2009.

[10] B. Green and T. Tao, "The primes contain arbitrarily long arithmetic progressions," *Annals of Mathematics*, vol. 167, pp. 481–547, 2008.

[11] B. Green and T. Tao, "Linear equations in primes," *Annals of Mathematics*, vol. 171, pp. 1753–1850, May 2010.

[12] L. Harn, C. C. Chang, and H. L. Wu, "An anonymous multi-receiver encryption based on RSA," *International Journal of Network Security*, vol. 15, no. 4, pp. 307–312, 2013.

[13] G. Ikorong, "A reformulation of the Goldbach conjecture," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 11, no. 4, pp. 465–469, 2008.

[14] Wolfram Research Inc, "Goldbach Conjecture," 2015. (http://mathworld.wolfram.com/GoldbachConjecture.html)

[15] H. W. Lenstra jr. and C. Pomerance, "Primality testing with Gaussian Periods," in *Proceedings of the 22nd Conference Kanpur on Foundations of Software Technology and Theoretical Computer Science (FST-TCS'02)*, vol. 2556, pp. 1, 2002.

[16] W. C. Lu, "Exceptional set of Goldbach number," *Journal of Number Theory*, vol. 130, pp. 2359–2392, Oct. 2010.

[17] I. Mikoss, "The prime numbers hidden symmetric structure and its relation to the twin prime infinitude and an improved prime number theorem," Technical Report MP-ARC-2006-314, 2006. (`http://www.ma.utexas.edu/mp_arc/c/06/06-314.pdf`)

[18] I. A. G. Nemron, "An original abstract over the twin primes, the Goldbach conjecture, the friendly numbers, the perfect numbers, the mersenne composite numbers, and the Sophie Germain primes," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 11, no. 6, pp. 715–726, 2008.

[19] Prime Number Patterns, "Prime number symmetry," 2010. (`http://primepatterns.wordpress.com/`)

[20] Z. Qin, *A Proof of the* G*oldbach's Conjecture*, The Economic Daily Press, China, Oct. 1995.

[21] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communincations of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[22] K. Slinker, "A proof of Goldbach's conjecture that all even numbers greater than four are the sum of two primes," Jan. 2008. (`http://arxiv.org/vc/arxiv/papers/0712/0712.2381v10.pdf`)

[23] Bo Yang, H. Ma, and S. Zhu, "A traitor tracing scheme based on the RSA system," *International Journal of Network Security*, vol. 5, no. 2, pp. 182–186, 2007.

[24] J. Ye and C. Liu, "A study of Goldbach's conjecture and Polignac's conjecture equivalence issues," *Cryptology ePrint Archive*, Report 2013/843, 2013. (`http://eprint.iacr.org/2013/843.pdf`)

[25] S. Zhang, "Goldbach conjecture and the least prime number in an arithmetic progression," *Comptes Rendus-Mathematique*, vol. 348, pp. 241–242, Mar. 2010.

[26] Y. Zhang, "Bounded gaps between primes," *Annals of Mathematics*, vol. 179, no. 3, pp. 1121-1174, 2014.

**Chenglian Liu** received his B.S degree in information management from National Union University (Taiwan) in 1992 and the MSc degree in national defense from National Defense University (Taiwan) in 2004. He studied his doctorate course at Royal Holloway, University of London from 2006 to 2009 under the supervised by Chris Mitchell. He is with a distinguished associate professor at Huizhou University since 2014. His research interests are in Key Agreement and Password Authentication, Number Theory and Cryptanalysis so on.

**Chin-Chen Chang** received his B.S. degree in applied mathematics in 1977 and the M.S. degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the college of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of the R.O.C. From 2002 to 2005, he was a Chair Professor of National Chung Cheng University. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures. He is a fellow of the IEEE.

**Zhi-Pan Wu** was born in 1975. He received his B.S degree in computer science at Xi-An University of Science and Technology in 1999, and received M.S. degree in software engineering at Central South University in 2006. He is with a senior lecturer at Huizhou University since 1999. His main research are image processing, computer vision and digital watermarking.

**Shi-Lin Ye** was born in 1992. He is an undergraduate of third grade student at Huizhou University currently. His main interests includes Network Security and Elementary Number Theory.

# Improvement of Green-Hohenberger Adaptive Oblivious Transfer: A Review

Zhengjun Cao[1], Lihua Liu[2]
*(Corresponding author: Zhengjun Cao)*

Department of Mathematics, Shanghai University[1]
No. 99, Shangda Road, Shanghai, China.
(Email: caozhj@shu.edu.cn)
Department of Mathematics, Shanghai Maritime University[2]
No. 1550, Haigang Ave, Pudong New District, Shanghai, China.

*(Received Mar. 20, 2015; revised and accepted May 2, 2015)*

## Abstract

In TCC'2011, Green and Hohenberger proposed an adaptive oblivious transfer (OT) scheme based on Decisional 3-Party Diffie-Hellman (3DDH) assumption. The encryption used in the scheme is a combination of Boneh-Boyen identity-based encryption and a variation of Hohenberger-Waters signature. The OT scheme is somewhat inefficient because it combines the two underlying schemes in a very simple way without making any optimizations. In this paper, we present a review on the Green-Hohenberger OT scheme and put forth a concrete improvement. We also show its security under 3DDH assumption. We think the optimizing skills developed in the paper are helpful for designing and analyzing other cryptographic schemes.

*Keywords: Adaptive oblivious transfer, redundant system parameters, 3-Party Diffie-Hellman assumption*

## 1 Introduction

The primitive of oblivious transfer (OT) introduced by Rabin [33] is of fundamental importance to secure multi-party computation [15, 37]. There are three main OT models: 1-out-of-2 oblivious transfer, 1-out-of-$n$ oblivious transfer and $k$-out-of-$n$ oblivious transfer. 1-out-of-2 oblivious transfer ($\mathrm{OT}_1^2$) as a generalization of Rabin's "oblivious transfer", was suggested by Even, Goldreich and Lempel [14]. In the model, the sender has two secrets $m_1$ and $m_2$ and would like to give the receiver one of them at the receiver's choice. Meanwhile, the receiver does not want the sender to know which secret he chooses. 1-out-of-$n$ oblivious transfer ($\mathrm{OT}_1^n$) is a generalization of $\mathrm{OT}_1^2$ proposed by Brassard et al. [5], in which the sender has $n$ secrets and want to give the receiver one of them at the receiver's choice. The receiver does not want the sender to know which secret he chooses. $k$-out-of-$n$ oblivious transfer ($\mathrm{OT}_k^n$) is a generalization of $\mathrm{OT}_1^n$, in which the sender has $n$ secrets and want to give the receiver $k$ of them at the receiver's choice. The receiver does not want the sender to know which secrets he chooses.

In an adaptive oblivious transfer, a sender commits to a database of messages and then repeatedly interacts with a receiver in such a way that the receiver obtains one message per interaction of his choice (and nothing more) while the sender learns nothing about any of the choices. In TCC'2011, Green and Hohenberger [18] presented an adaptive OT scheme based on 3DDH assumption which says that given $(g, g^a, g^b, g^c, Q)$ where $g$ generates a bilinear group of prime order $p$ and $a, b, c$ are selected randomly from $\mathbb{Z}_p$, it is hard to decide if $Q = g^{abc}$. In their scheme, the sender commits to a database of $n$ messages by publishing an encryption of each message and a signature on each encryption. Then, each transfer phase can be executed in time independent of $n$ as the receiver blinds one of the encryptions and proves knowledge of the blinding factors and a signature on this encryption, after which the sender helps the receiver decrypt the chosen ciphertext.

The encryption used in the scheme is a combination of Boneh-Boyen IBE scheme [3] and a variation of Hohenberger-Waters signature [19]. However, it combines the two underlying schemes in a very simple way without making any optimizations. Concretely, there are two drawbacks:

1) It sets the secret key as $(a, b)$, where $a$ is used only for decryption and $b$ is used only for signing, separately. But we know it is usual that a single secret key $a$ can be used simultaneously for both signing and decryption.

2) For random $r, s, t \in \mathbb{Z}_p$, it expresses the ciphertext as

$$\Big(g^r,\ (g_1^j h)^r,\ M \cdot e(g_1, g_2)^r,\ g^t,\ (u^r v^s d)^b (g_3^j h)^t,\ u^r,\ s\Big)$$

where $p, g, e(\cdot,\cdot), g_1, g_2, g_3, u, v, d, h$ are included in public parameters. The session key $s$ is directly exposed. That means the corresponding parameter $v$ might be removed reasonably.

In this paper, we present an improvement of Green-Hohenberger adaptive OT scheme and show its security under 3DDH assumption. We also correct some typos in the original scheme. The analysis and optimizing skills presented in the paper is novel. We think they are helpful for optimizing other cryptographic schemes.

### 1.1 Related Works

In past decades, there were many works on the research of $\mathrm{OT}_k^n$, such as Bellare and Micali [1], Naor and Pinkas [30, 31, 32], Mu, Zhang, and Varadharajan [29], Chu and Tzeng [12]. Recently, Chang and Lai [10], Chang and Lee [11], and Liu et al. [2, 13, 20, 22, 26, 27, 28, 35, 36, 38] have presented some efficient $\mathrm{OT}_k^n$ schemes.

Naor and Pinkas [31] initiated the study on the problem of oblivious transfer with adaptive queries. Their work was followed by [6, 12, 16, 18, 24, 25, 39]. The Camenisch-Neven-Shelat OT scheme [6] uses bilinear groups as the building block and adopts the paradigm of "encryption and proof of knowledge" to force the sender to keep the consistency of the transferred messages. The paradigm has been used in the latter OT protocols [16, 18, 24, 25, 39]. In Asiacrypt'08, Green and Hohenberger [17] presented a universally composable adaptive oblivious transfer scheme which makes use of a signature built from the Boneh-Boyen IBE [3]. Recently, Cao, Lafitte and Markowitch [9] have shown that the signature scheme was selectively forgeable and the reduction used in their proof was flawed. Cao and Cao [8] has improved Camenisch-Neven-Shelat OT scheme and reaffirmed that the transferred messages in any OT scheme must be recognizable to the receiver. Otherwise, the receiver cannot decide which message should to be extracted. The gist of the primitive of OT has been really neglected for a long time. It is a big step towards the practical use of OT.

### 1.2 The Definition of Adaptive k-out-of-N Oblivious Transfer

The definition can be found in [18]. For completeness, we now describe it as follows. An adaptive oblivious transfer scheme is a tuple of algorithms $(\mathsf{S}_\mathsf{I}, \mathsf{R}_\mathsf{I}, \mathsf{S}_\mathsf{T}, \mathsf{R}_\mathsf{T})$. During the initialization phase, the Sender and the Receiver conduct an interactive protocol, where the Sender runs $\mathsf{S}_\mathsf{I}(M_1, \cdots, M_N)$ to obtain state value $S_0$, and the Receiver runs $\mathsf{R}_\mathsf{I}()$ to obtain state value $R_0$. Next, for $1 \le i \le k$, the $i^{th}$ transfer proceeds as follows: the Sender runs $\mathsf{S}_\mathsf{T}(S_{i-1})$ to obtain state value $S_i$, and the Receiver runs $\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)$ where $1 \le \sigma_i \le N$ is the index of the message to be received. The Receiver obtains state information $R_i$ and the message $M'_{\sigma_i}$ or $\perp$ indicating failure. To define the Sender and Receiver security, we need the following experiments.

**Real Experiment.** In the experiment of $\mathbf{Real}_{\hat{\mathsf{S}},\hat{\mathsf{R}}}$ $(N, k, M_1, \cdots, M_N, \Sigma)$, the possibly cheating sender $\hat{\mathsf{S}}$ is given messages $(M_1, \cdots, M_N)$ as input and interacts with the possibly cheating receiver $\hat{\mathsf{R}}(\Sigma)$, where $\Sigma$ is a selection algorithm that on input the full collection of messages thus far received, outputs the index $\sigma_i$ of the next message to be queried. At the beginning of the experiment, both $\hat{\mathsf{S}}$ and $\hat{\mathsf{R}}$ output initial states $(S_0, R_0)$. In the transfer phase, for $1 \le i \le k$ the sender computes $S_i \leftarrow \hat{\mathsf{S}}(S_{i-1})$, and the receiver computes $(R_i, M'_i) \leftarrow \hat{\mathsf{R}}(R_{i-1})$, where $M'_i$ may or may not be equal to $M_i$. At the end of the $k$-th transfer the output of the experiment is $(S_k, R_k)$.

**Ideal Experiment.** In the experiment of $\mathbf{Ideal}_{\hat{\mathsf{S}}',\hat{\mathsf{R}}'}$ $(N, k, M_1, \cdots, M_N, \Sigma)$ the possibly cheating sender algorithm $\hat{\mathsf{S}}'$ generates messages $(M_1^*, \cdots, M_N^*)$ and transmits them to a trusted party $\mathsf{T}$. In the $i$-th round $\hat{\mathsf{S}}'$ sends a bit $b_i$ to $\mathsf{T}$; the possibly cheating receiver $\hat{\mathsf{R}}'(\Sigma)$ transmits $\sigma_i^*$ to $\mathsf{T}$. If $b_i = 1$ and $\sigma_i^* \in \{1, \cdots, N\}$ then $\mathsf{T}$ hands $M^*_{\sigma_i^*}$ to $\hat{\mathsf{R}}'$. If $b_i = 0$ then $\mathsf{T}$ hands $\perp$ to $\hat{\mathsf{R}}'$. After the $k$-th transfer the output of the experiment is $(S_k, R_k)$.

**Sender Security.** An $\mathrm{OT}^N_{k\times 1}$ provides Sender security if for every real-world p.p.t. receiver $\hat{\mathsf{R}}$ there exists a p.p.t. ideal-world receiver $\hat{\mathsf{R}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \cdots, M_N)$, $\Sigma$, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \cdots, M_N, \Sigma)$$

$$\overset{c}{\approx} \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \cdots, M_N, \Sigma),$$

where $\ell(\cdot)$ is a polynomially-bounded function.

**Receiver Security.** An $\mathrm{OT}^N_{k\times 1}$ provides Receiver security if for every real-world p.p.t. sender $\hat{\mathsf{S}}$ there exists a p.p.t. ideal-world sender $\hat{\mathsf{S}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \cdots, M_N)$, $\Sigma$, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \cdots, M_N, \Sigma)$$

$$\overset{c}{\approx} \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \cdots, M_N, \Sigma).$$

## 2 A Simple Security Assumption

Let *BMsetup* be an algorithm that, on input $1^\kappa$, outputs the parameters for a bilinear mapping as $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e)$, where $g$ generates $\mathbb{G}$, the groups $\mathbb{G}$ and $\mathbb{G}_T$ have prime order $p$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. It is both:

(*bilinear*) for all $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$,

$$e(g^a, g^b) = e(g, g)^{ab};$$

(*non-degenerate*) if $g$ generates $\mathbb{G}$, then $e(g, g) \neq 1$.

**Assumption 1.** *(Decisional 3-Party Diffie-Hellman (3DDH))[4] Let $g$ generate a group $\mathbb{G}$ of prime order $p \in \Theta(2^{\lambda})$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is 1/2 plus an amount negligible in $\lambda$:*

$$Pr[g, z_0 \leftarrow \mathbb{G}; a, b, c \leftarrow \mathbb{Z}_p;\ z_1 \leftarrow g^{abc};$$

$$d \leftarrow \{0,1\}; d' \leftarrow \mathcal{A}(g, g^a, g^b, g^c, z_d) : d = d'].$$

We use the notation of Camenisch and Stadler [7] for the proofs of knowledge. For instance, $ZKPoK\{(x,h) : y = g^x \wedge H = e(y,h) \wedge (1 \leq x \leq n)\}$ denotes a zero-knowledge proof of knowledge of an integer $x$ and a group element $h \in \mathbb{G}$ such that $y = g^x$ and $H = e(y,h)$ holds and $1 \leq x \leq n$. All values not enclosed in ()'s are assumed to be known to the verifier.

# 3 Review and Analysis of Green-Hohenberger Adaptive OT

## 3.1 Review

This protocol follows the assisted (or blind) decryption paradigm [6, 17, 21]. The Sender begins the OT protocol by encrypting each message in the database and publishing these values to the Receiver. The Receiver then checks that each ciphertext is well-formed. See the following Table 1 for details.

**Ciphertext Structure.** The Sender's public parameters $pk$ include $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e)$ and generators $(g_1, g_2, h, g_3, g_4, u, v, d) \in \mathbb{G}^8$. For message $M \in \mathbb{G}_T$, identity $j \in \mathbb{Z}_p$, and random values $r, s, t \in \mathbb{Z}_p$, the ciphertext is expressed as:

$$C = \left(g^r, (g_1^j h)^r, M \cdot e(g_1, g_2)^r, g^t, (u^r v^s d)^b (g_3^j h)^t, u^r, s\right).$$

Given only $pk, j$, the VerifyCiphertext function validates that the ciphertext has this structure.

**VerifyCiphertext** $(pk, C, j)$**.** Parse $C$ as $(c_1, \cdots, c_7)$ and $pk$ to obtain $g, g_1, h, g_3, g_4, u, v, d$. This routine outputs 1 if and only if the following equalities hold:

$$\begin{aligned} e(g_1^j h, c_1) &= e(g, c_2)\ \wedge\ e(g, c_6) \\ &= e(c_1, u) \wedge e(g, c_5) \\ &= e(g_4, c_6 v^{c_7} d) e(c_4, g_3^j h). \end{aligned}$$

## 3.2 Drawbacks

The encryption used in the scheme is a combination of the Boneh-Boyen IBE scheme [3] and a variation of the Hohenberger-Waters signature [19]. It combines the two base schemes in a very simple way. Concretely, there are three drawbacks:

1) It sets the secret key as $(a, b)$, where $a$ is used only for decryption and $b$ is used only for signing, separately. But it is usual that a single secret key $a$ can be simultaneously used for both signing and decryption. We will set $b = a$ and show that the setting does not endanger its security. That means the generator $g_4$ could be removed.

2) For random $r, s, t \in \mathbb{Z}_p$, it expresses the ciphertext as

$$\left(g^r,\ (g_1^j h)^r,\ M \cdot e(g_1, g_2)^r, g^t,\ (u^r v^s d)^b (g_3^j h)^t,\ u^r,\ s\right)$$

Notice that the session key $s$ is *directly exposed*. That means the generator $v$ could be removed, too. The redundant setting is due to that the authors follow the Hohenberger-Waters signature based on RSA assumption (see Section 3 in [19]), which does require a Chameleon hash function. We would like to stress that the structure $u^M v^s$ in a bilinear group $\mathbb{G}$ has no the special property of a chameleon hash function because one can not find $s'$ satisfying $u^M v^s = u^{M'} v^{s'}$, given $M, M'$ and $s$, where $u, v$ are two random elements of $\mathbb{G}$. The authors misapplied the structure.

3) The generator $g_2$ is used only for the blind decryption and the generator $g_3$ is used only for the VerifyCiphertext. For simplicity, we could explicitly set that $g_3 = g_2$. That is to say, the generator $g_3$ might be redundant. By the way, the generator $d$ is required necessarily for the Hohenberger-Waters signature based on CDH assumption [19]. The generator $h$ facilitates the security proof of the Hohenberger-Waters signature. If $d$ is removed, then we have the following attack. Given a valid ciphertext

$$\begin{aligned} C &= (c_1, \cdots, c_7) \\ &= (g^r, (g_1^j h)^r, M \cdot e(g_1, g_2)^r, g^t, \\ &\qquad (u^r v^s)^b (g_3^j h)^t, u^r, s). \end{aligned} \tag{1}$$

An adversary can take a random $\theta \in \mathbb{Z}_p$ and compute

$$\begin{aligned} \hat{C} &= (\hat{c}_1, \cdots, \hat{c}_7) \\ &= (g^{r\theta}, (g_1^j h)^{r\theta}, M^{\theta} \cdot e(g_1, g_2)^{r\theta}, g^{t\theta}, \\ &\qquad \left((u^r v^s)^b (g_3^j h)^t\right)^{\theta}, u^{r\theta}, s\theta). \end{aligned} \tag{2}$$

The ciphertext $\hat{C}$ is valid because

$$\begin{aligned} e(g_1^j h, \hat{c}_1) &= e(g, \hat{c}_2)\ \wedge\ e(g, \hat{c}_6) \\ &= e(\hat{c}_1, u) \wedge e(g, \hat{c}_5) \\ &= e(g_4, \hat{c}_6 v^{\hat{c}_7}) e(\hat{c}_4, g_3^j h). \end{aligned}$$

*Remark.* The random $y \in \mathbb{Z}_p$ chosen by the receiver is not used at all. This is a typo.

Table 1: The Green-Hohenberger adaptive OT scheme

| $\mathsf{S}_\mathsf{I}(M_1, \cdots, M_N)$ | $\mathsf{R}_\mathsf{I}()$ |
|---|---|
| 1. Select $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow$ BMsetup $(1^\kappa)$,<br>$a, b \leftarrow \mathbb{Z}_p$, $g_2, g_3, h, u, v, d \leftarrow \mathbb{G}$<br>and set $g_1 \leftarrow g^a, g_4 \leftarrow g^b$.<br>$pk \leftarrow (\gamma, g_1, g_2, g_3, g_4, h, u, v, d)$, $sk \leftarrow (a, b)$.<br>2. For $j = 1$ to $N$, select $r_j, s_j, t_j \leftarrow \mathbb{Z}_p$<br>and set: $C_j \leftarrow [g^{r_j}, (g_1^j h)^{r_j}, M_j e(g_1, g_2)^{r_j},$<br>$g^{t_j}, (u^{r_j} v^{s_j} d)^b (g_3^j h)^{t_j}, u^{r_j}, s_j]$<br>3. Send $(pk, C_1, \cdots, C_N)$ to Receiver.<br>4. Conduct $ZKPoK\{(a) : g_1 = g^a\}$.<br>Output $S_0 = (pk, sk)$. | 5. Verify $pk$ and the proof.<br>Check for $j = 1$ to $N$:<br>VerifyCiphertext $(pk, C_j, j)$=1.<br>If any check fails, output $\perp$.<br>Output $R_0 = (pk, C_1, \cdots, C_N)$. |
| $\mathsf{S}_\mathsf{T}(S_{i-1})$ | $\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)$ |
| 3. Set $R = e(v_1, g_2^a)$.<br>4. Send $R$ to Receiver and conduct:<br>$ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.<br>Output $S_i = S_{i-1}$. | 1. Parse $C_{\sigma_i}$ as $(c_1, \cdots, c_7)$,<br>select $\underline{x, y \leftarrow \mathbb{Z}_p}$<br>and compute $v_1 = g^x c_1$.<br>2. Send $v_1$ to Sender, and conduct:<br>$WIPoK\{(\sigma_i, x, c_2, c_4, c_5, c_6, c_7) :$<br>$e(v_1/g^x, (g_1^{\sigma_i} h)) = e(c_2, g) \wedge$<br>$e(c_6, g) = e(v_1/g^x, u) \wedge$<br>$e(c_5, g) = e(c_6 v^{c_7} d, g_4) e(c_4, g_3^{\sigma_i} h)\}$<br>5. If the proof does not verify, output $\perp$.<br>Else output $M'_{\sigma_i} = \frac{c_3 \cdot e(g_1, g_2)^x}{R}$.<br>Output $R_i = (R_{i-1}, M'_{\sigma_i})$ |

# 4 An Improvement of Green-Hohenberger OT Scheme and Its Security Proof

## 4.1 The Improvement

The improvement is obtained by removing the redundant generators $g_3, g_4, v$. See the Table 2 for details.

**Ciphertext Structure.** The Sender's public parameters $pk$ include $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e)$ and generators $(g_1, g_2, h, u, d) \in \mathbb{G}^5$. For message $M \in \mathbb{G}_T$, identity $j \in \mathbb{Z}_p$, and random values $r, t \in \mathbb{Z}_p$, the ciphertext is expressed as:

$$\left(g^r, (g_1^j h)^r, M \cdot e(g_1, g_2)^r, g^t, (u^r d)^a (g_2^j h)^t, u^r\right).$$

Given only $pk, j$, the VerifyCiphertext function validates that the ciphertext has this structure.

**VerifyCiphertext** $(pk, C, j)$**.** Parse $C$ as $(c_1, \cdots, c_6)$ and $pk$ to obtain $g, g_1, g_2, h, u, d$. This routine outputs 1 if and only if the following equalities hold:

$$\begin{aligned} e(g_1^j h, c_1) &= e(g, c_2) \ \wedge \ e(g, c_6) \\ &= e(c_1, u) \wedge e(g, c_5) \\ &= e(g_1, c_6 d) e(c_4, g_2^j h). \end{aligned}$$

**Correctness.**

$$\begin{aligned} e(g_1^j h, c_1) &= e(g_1^j h, g^{r_j}) = e((g_1^j h)^{r_j}, g) \\ &= e(g, c_2) \\ e(g, c_6) &= e(g, u^{r_j}) = e(g^{r_j}, u) = e(c_1, u) \\ e(g, c_5) &= e\left(g, (u^{r_j} d)^a (g_2^j h)^{t_j}\right) \\ &= e\left(g, (u^{r_j} d)^a\right) e\left(g, (g_2^j h)^{t_j}\right) \\ &= e(g_1, c_6 d) e(c_4, g_2^j h) \\ \frac{c_3 \cdot e(g_1, g_2)^x}{R} &= \frac{M_j\, e(g_1, g_2)^{r_j} \cdot e(g_1, g_2)^x}{e(g^x c_1, g_2^a)} \\ &= \frac{M_j\, e(g_1, g_2)^{r_j} \cdot e(g_1, g_2)^x}{e(g^x, g_2^a) e(g^{r_j}, g_2^a)} = M_j \end{aligned}$$

## 4.2 Security Proof

The improvement is sender-secure and receiver-secure in the full simulation model under 3DDH assumption. The security proof is somewhat like that of the original scheme [18]. For completeness, we now describe it as follows.

Sender security. Given a (possibly cheating) real-world receiver $\hat{\mathsf{R}}$, we show how to construct an ideal-world receiver $\hat{\mathsf{R}}'$ such that all p.p.t. distinguishers have at most negligible advantage in distinguishing the distribution of an honest real-world sender $\mathsf{S}$ interacting with $\hat{\mathsf{R}}$ ($\mathrm{Real}_{\mathsf{S},\hat{\mathsf{R}}}$)

Table 2: The improvement of Green-Hohenberger adaptive OT scheme

| $\mathsf{S_I}(M_1, \cdots, M_N)$ | $\mathsf{R_I}()$ |
|---|---|
| 1. Select $\gamma = (p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow$ BMsetup $(1^\kappa)$,<br>$a \leftarrow \mathbb{Z}_p$, choose $g_2, h, u, d \leftarrow \mathbb{G}$<br>and set $g_1 \leftarrow g^a$.<br>$pk \leftarrow (\gamma, g_1, g_2, h, u, d)$, $sk \leftarrow a$.<br>2. For $j = 1$ to $N$, select $r_j, t_j \leftarrow \mathbb{Z}_p$<br>and set: $C_j \leftarrow [g^{r_j}, (g_1^j h)^{r_j}, M_j\, e(g_1, g_2)^{r_j},$<br>$g^{t_j}, (u^{r_j} d)^a (g_2^j h)^{t_j}, u^{r_j}]$<br>3. Send $(pk, C_1, \cdots, C_N)$ to Receiver.<br>4. Conduct $ZKPoK\{(a) : g_1 = g^a\}$.<br><br>Output $S_0 = (pk, sk)$. | 5. Verify $pk$ and the proof.<br>Check for $j = 1$ to $N$:<br>$\mathsf{VerifyCiphertext}$ $(pk, C_j, j)$=1.<br>If any check fails, output $\perp$.<br><br>Output $R_0 = (pk, C_1, \cdots, C_N)$. |
| $\mathsf{S_T}(S_{i-1})$<br><br>3. Set $R = e(v_1, g_2^a)$.<br>4. Send $R$ to Receiver and conduct:<br>$ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}$.<br><br>Output $S_i = S_{i-1}$. | $\mathsf{R_T}(R_{i-1}, \sigma_i)$<br>1. Parse $C_{\sigma_i}$ as $(c_1, \cdots, c_6)$, select $x \leftarrow \mathbb{Z}_p$<br>and compute $v_1 = g^x c_1$.<br>2. Send $v_1$ to Sender, and conduct:<br>$WIPoK\{(\sigma_i, x, c_2, c_4, c_5, c_6) :$<br>$e(v_1/g^x, (g_1^{\sigma_i} h)) = e(c_2, g) \wedge$<br>$e(c_6, g) = e(v_1/g^x, u) \wedge$<br>$e(c_5, g) = e(c_6 d, g_1) e(c_4, g_2^{\sigma_i} h)\}$<br><br>5. If the proof does not verify, output $\perp$.<br>Else output $M'_{\sigma_i} = \frac{c_3 \cdot e(g_1, g_2)^x}{R}$.<br><br>Output $R_i = (R_{i-1}, M'_{\sigma_i})$ |

from that of $\hat{\mathsf{R}}'$ interacting with the honest ideal-world sender $\mathsf{S}'$ (Ideal$_{\mathsf{S}',\hat{\mathsf{R}}'}$ ).

1) To begin, $\hat{\mathsf{R}}'$ selects a random collection of messages $\bar{M}_1, \cdots, \bar{M}_N \leftarrow \mathbb{G}_T$ and follows the $\mathsf{S_I}$ algorithm with these as input up to the point where it obtains $(pk, C_1, \cdots, C_N)$.

2) It sends $(pk, C_1, \cdots, C_N)$ to $\hat{\mathsf{R}}$ and then simulates the interactive proof

$$ZKPoK\{(a) : g_1 = g^a\}.$$

(Even though $\hat{\mathsf{R}}'$ knows $sk = a$, it ignores this value and simulate this proof step.)

3) For each of $k$ transfers initiated by $\hat{\mathsf{R}}$,

   a. $\hat{\mathsf{R}}'$ verifies the received WIPoK and uses the knowledge extractor $\mathsf{E}_2$ to obtain the values $\sigma_i, x, c_1, c_2, c_3, c_4$ from it. $\hat{\mathsf{R}}'$ aborts and outputs $\mathsf{error}$ when $\mathsf{E}_2$ fails.

   b. When $\sigma_i \in [1, N]$, $\hat{\mathsf{R}}'$ queries the trusted party $\mathsf{T}$ to obtain $M_{\sigma_i}$, parses $C_{\sigma_i}$ as $(c_1, \cdots, c_6)$ and responds with

$$R = \frac{c_3\, e(g_1, g_2)^x}{M_{\sigma_i}}$$

(if $\mathsf{T}$ returns $\perp$, $\hat{\mathsf{R}}'$ aborts the transfer). When $\sigma_i \notin [1, N]$, $\hat{\mathsf{R}}'$ follows the normal protocol. In both cases, $\hat{\mathsf{R}}'$ simulates

$$ZKPoK\{(a) : R = e(v_1, g_2^a) \wedge g_1 = g^a\}.$$

4) $\hat{\mathsf{R}}'$ uses $\hat{\mathsf{R}}$'s output as its own.

**Theorem 1.** *Let $\epsilon_{ZK}$ be the maximum advantage with which any p.p.t. algorithm distinguishes a simulated ZKPoK, and $\epsilon_{Ext}$ be the maximum probability that the extractor $\mathsf{E}_2$ fails (with $\epsilon_{ZK}$ and $\epsilon_{Ext}$ both negligible in $\kappa$). If all p.p.t. algorithms have negligible advantage $\leq \epsilon$ at solving the 3DDH problem, then:*

$$Pr\Big[D(Real_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \cdots, M_N, \Sigma)) = 1\Big] -$$

$$Pr\Big[D(Ideal_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \cdots, M_N, \Sigma)) = 1\Big] \leq$$

$$(k+1)\epsilon_{ZK} + k\epsilon_{Ext} + N\epsilon\left(1 + \frac{p}{p-1}\right).$$

*Proof.* We first define the following games:

**Game 0.** The real-world experiment conducted between $\mathsf{S}$ and $\hat{\mathsf{R}}$ (Real$_{\mathsf{S},\hat{\mathsf{R}}}$).

**Game 1.** This game modifies **Game 0** as follows: (1) each of $\mathsf{S}$'s ZKPoK executions is replaced with a simulated proof of the same statement, and (2) the knowledge extractor $\mathsf{E}_2$ is used to obtain the values $(\sigma_i, x, \bar{c}_4, \bar{c}_5, \bar{c}_6)$ from each of $\hat{\mathsf{R}}$'s transfer queries. Whenever the extractor fails, $\mathsf{S}$ terminates the experiment and outputs the distinguished symbol $\mathsf{error}$.

(*There is a typo in the original argument. It says that "the knowledge extractor* $\mathsf{E}_2$ *is used to obtain the values* $(\sigma_i, x, y, z, \bar{c}_4, \bar{c}_5, \bar{c}_6, \bar{c}_7)$ *from each of* $\hat{R}$*'s transfer queries". We stress that both the values* $y, z$ *are not used at all.*)

**Game 2.** This game modifies **Game 1** such that, whenever the extracted value $\sigma_i \in [1, N]$, $\mathsf{S}$'s response $R$ is computed using the following approach: parse $C_{\sigma_i} = (c_1, \cdots, c_6)$ and set

$$R = \frac{c_3\, e(g_1, g_2)^x}{M_{\sigma_i}}.$$

When $\sigma_i \notin [1, N]$, the response is computed using the normal protocol.

**Game 3.** This game modifies **Game 2** by replacing the input to $\mathsf{S_I}$ with a dummy vector of random messages $\bar{M}_1, \cdots, \bar{M}_N \in \mathbb{G}_T$. However when $\mathsf{S}$ computes a response value using the technique of **Game 2**, the response is based on the original message vector $M_1, \cdots, M_N$. We claim that the distribution of this game is equivalent to that of $\mathrm{Ideal}_{\mathsf{S}', \hat{\mathsf{R}}'}$.

For notational convenience, define:

$$\mathrm{Adv}[\text{Game i}] = \Pr[D(\text{Game i}) = 1] - \Pr[D(\text{Game 0}) = 1].$$

By the following Lemmas, we then obtain

$$\mathrm{Adv}[\text{Game 3}] \le (k+1)\epsilon_{ZK} + k\epsilon_{Ext} + N\epsilon(1 + \frac{p}{p-1}).$$

□

**Lemma 1.** *If all p.p.t. algorithms D distinguish a simulated ZKPoK with advantage at most* $\epsilon_{ZK}$ *and the extractor* $\mathsf{E}_2$ *fails with probability at most* $\epsilon_{Ext}$*, then* $Adv[Game\ 1] \le (k+1)\epsilon_{ZK} + k\epsilon_{Ext}$.

*Proof.* See the proof of Lemma A.1 in [18]. □

**Lemma 2.** *If no p.p.t. algorithm has advantage* $> \epsilon$ *in solving the 3DDH problem, then*

$$Adv[Game\ 2] - Adv[Game\ 1] \le \frac{Np}{p-1} \cdot \epsilon$$

*Proof.* For every query where $\sigma_i \notin [1, N]$, $\mathsf{S}$ calculates the response $R$ as in the normal protocol, and thus the distribution of $R$ is identical to **Game 1**. Thus we need only consider queries where $\sigma_i \in [1, N]$.

Given a transfer request containing $v_1$, let us implicitly define

$$g^{r'} = v_1 / g^x$$

for some $r' \in \mathbb{Z}_p$. Express the $\sigma_i$-th ciphertext in the database as $C_{\sigma_i} = (c_1, \cdots, c_6)$. If $g^{r'} = c_1$ then the computed response $R$ will have the same distribution as in the normal protocol. To show this, let $c_1 = g^{r_{\sigma_i}}$ for some $r_{\sigma_i} \in \mathbb{Z}_p$ and $c_3 / M_{\sigma_i} = e(g_1, g_2)^{r_{\sigma_i}}$. We can now write the normal calculation of $R$ as:

$$\begin{aligned} R &= e(c_1 g^x, g_2^a) = e(g^{r_{\sigma_i}} g^x, g_2^a) \\ &= e(g_1, g_2)^{r_{\sigma_i}} e(g_1, g_2)^x = \frac{c_3\, e(g_1, g_2)^x}{M_{\sigma_i}}. \end{aligned}$$

It remains only to consider the case where $g^{r'} \neq c_1$. We will refer to this as a *forged query* and argue that $\hat{\mathsf{R}}$ cannot issue such a query except with negligible probability under the 3DDH assumption in $\mathbb{G}$. Specifically, if $\hat{\mathsf{R}}$ submits a forged query with non-negligible probability, then we can construct a solver $\mathcal{B}$ for 3DDH that succeeds with non-negligible advantage.

We now describe the solver $\mathcal{B}$. $\mathcal{B}$ takes as input a 3DDH tuple $(g, g^\tau, g^\psi, g^\omega, Z)$, where $Z = g^{\tau\psi\omega}$ or is random, and each value $\tau, \varphi, \omega$ was chosen at random from $\mathbb{Z}_p$. It will simulate $\mathsf{S}$'s interaction with $\hat{\mathsf{R}}$ via the following simulation.

**Simulation Setup.** $\mathcal{B}$ first picks $j^* \leftarrow [1, N]$ and $y_d, x_d, x_h, x_z \leftarrow \mathbb{Z}_p$. It sets

$$u = g^\psi, d = g^{-\psi x_d} g^{y_d}, h = g^{-\psi j^*} g^{x_h}, g_2 = g^\psi g^{x_z}, g_1 = g^\tau.$$

Thus, we implicitly have $a = \tau$. The remaining components of $pk$ are chosen as in the real protocol.
(*There is a typo in the original argument. It says that "*$\mathcal{B}$ *first picks* $j^* \leftarrow [1, N]$ *and* $a, y_v, y_d, x_v, x_d, x_h, x_z, r_j, t_j \leftarrow \mathbb{Z}_p$*". Clearly, the secret key a for decryption is not known to the solver* $\mathcal{B}$*. Besides, it is not necessary for* $\mathcal{B}$ *to pick* $r_j, t_j$ *in the* Setup *because they are not used at all in the phase.*)

For $j = 1$ to $N$, $\mathcal{B}$ generates each correctly-distributed ciphertext $C_j = (c_1, \cdots, c_6)$ as follows:

**The simulation for** $j = j^*$**.** Pick $t_j \leftarrow \mathbb{Z}_p$ and set the ciphertext as:

$$(c_1, \cdots, c_6) = \Big(g^{x_d}, (g_1^j h)^{x_d}, M \cdot e(g_1, g_2)^{x_d}, \\ g^{t_j}, (g^\tau)^{y_d} (g_2^j h)^{t_j}, u^{x_d}\Big).$$

The ciphertext is well-formed because:

$$\begin{aligned} e(g_1^j h, c_1) &= e(g_1^j h, g^{x_d}) = e((g_1^j h)^{x_d}, g) = e(g, c_2) \\ e(g, c_6) &= e(g, u^{x_d}) = e(g^{x_d}, u) = e(c_1, u) \\ e(g, c_5) &= e\left(g, (g^\tau)^{y_d} (g_2^j h)^{t_j}\right) \\ &= e\left(g, (u^{x_d} d)^\tau\right) e\left(g, (g_2^j h)^{t_j}\right) \\ &= e(g_1, c_6 d) e(c_4, g_2^j h). \end{aligned}$$

**The simulation for** $j \neq j^*$**.** Pick $r_j, t'_j \leftarrow \mathbb{Z}_p$. Set

$$Y = g^{t'_j} / (g^\tau)^{(r_j - x_d)/(j - j^*)}$$

and the ciphertext as:

$$(c_1, \cdots, c_6) = \Big(g^{r_j}, (g_1^j h)^{r_j}, M \cdot e(g_1, g_2)^{r_j}, Y,$$

$$(g^{\tau})^{y_d} \cdot Y^{x_z j + x_h} \cdot (g^{\psi})^{t'_j (j-j^*)}, u^{r_j} \Big).$$

Let us define $Y = g^{t_j}$ and thus implicitly set

$$t_j = t'_j - \tau(r_j - x_d)/(j - j^*),$$

which is randomly distributed in $\mathbb{Z}_p$. Just by inspection, it's clear that all of the elements except $c_5$ are correctly distributed. Thus it remains to show that:

$$(g^{\tau})^{y_d} \cdot Y^{x_z j + x_h} \cdot (g^{\psi})^{t'_j (j-j^*)} = (u^{r_j} d)^{\tau} (g_2^j h)^{t_j}$$

In fact, we have:

$$\begin{aligned}
c_5 &= (g^{\tau})^{y_d} \cdot Y^{x_z j + x_h} \cdot (g^{\psi})^{t'_j (j-j^*)} \\
&= (g^{\tau})^{y_d} \cdot (g^{t_j})^{x_z j + x_h} \cdot (g^{\psi})^{t'_j (j-j^*)} \\
&= (g^{\tau\psi})^{r_j - x_d} (g^{\tau})^{y_d} \cdot (g^{t_j})^{x_z j + x_h} \\
&\qquad \cdot (g^{\psi})^{t'_j (j-j^*)} (g^{-\tau\psi})^{r_j - x_d} \\
&= (g^{\psi(r_j - x_d)})^{\tau} (g^{y_d})^{\tau} \cdot (g^{x_z j + x_h})^{t_j} \\
&\qquad \cdot (g^{\psi})^{t'_j (j-j^*)} (g^{-\tau\psi})^{r_j - x_d} \\
&= ((g^{\psi r_j})(g^{-\psi x_d + y_d}))^{\tau} \cdot (g^{x_z j + x_h})^{t_j} \\
&\qquad \cdot (g^{\psi})^{t'_j (j-j^*)} (g^{-\tau\psi})^{r_j - x_d} \\
&= (u^{r_j} d)^{\tau} \cdot (g^{x_z j + x_h})^{t_j} \cdot (g^{\psi})^{t'_j (j-j^*)} (g^{-\tau\psi})^{r_j - x_d} \\
&= (u^{r_j} d)^{\tau} \cdot (g^{x_z j + x_h})^{t_j} \cdot (g^{\psi(j-j^*)})^{t'_j - \tau(r_j - x_d)/(j-j^*)} \\
&= (u^{r_j} d)^{\tau} \cdot (g^{x_z j + x_h})^{t_j} \cdot (g^{\psi(j-j^*)})^{t_j} \\
&= (u^{r_j} d)^{\tau} \cdot ((g^{\psi + x_z})^j g^{-\psi j^* + x_h})^{t_j} \\
&= (u^{r_j} d)^{\tau} \cdot (g_2^j h)^{t_j}.
\end{aligned}$$

**Answering Queries.** Upon receiving a query from $\hat{\mathsf{R}}$, $\mathcal{B}$ verifies the accompanying WIPoK and extracts $(\sigma_i, x, \bar{c}_4, \bar{c}_5, \bar{c}_6)$ and the value $v_1$. Note that $\hat{\mathsf{R}}$ must issue at least one forged query where $v_1/g^x$ is not equal to the first element of $C_{\sigma_i}$ . When this occurs, if $\sigma_i \neq j^*$ then $\mathcal{B}$ aborts and outputs a random bit.

Otherwise let us consider the distribution of $\hat{\mathsf{R}}$'s query. For some $t, r' \in \mathbb{Z}_p$ the soundness of the WIPoK ensures that

$$(v_1/g^x, \bar{c}_6) = (g^{r'}, u^{r'})$$

and

$$(\bar{c}_4, \bar{c}_5) = (g^t, (u^{r'} d)^a (g_2^{\sigma_i} h)^t).$$

By substitution we obtain:

$$\begin{aligned}
\bar{c}_5 &= (g^{\psi r'} g^{-\psi x_d + y_d})^{\tau} (g^{(\psi + x_z) j^*} g^{-\psi j^*} g^{x_h})^t \\
&= g^{\tau\psi(r' - x_d)} g^{\tau y_d} g^{t(x_z j^* + x_h)}.
\end{aligned}$$

Let us implicitly define the value

$$h' = (v_1/g^x) g^{-x_d} = g^{r' - x_d}.$$

$\mathcal{B}$ can obtain $h'^{\tau\psi}$ by computing

$$\bar{c}_5 / (g^{\tau y_d} \bar{c}_4^{x_z j^* + x_h}).$$

Provided that $h' \neq 1$, $\mathcal{B}$ can now compute a solution to the 3DDH problem by comparing

$$e(h'^{\tau\psi}, g^{\omega}) \overset{?}{=} e(Z, h').$$

If $h' = 1$ then $\mathcal{B}$ aborts and outputs a random bit.

*Probability of abort.* There are two conditions in which $\mathcal{B}$ aborts: (1) when $\hat{\mathsf{R}}$ does not issue a forgery for $\sigma_i = j^*$, and (2) when $\sigma_i = j^*$ but $(v_1/g^x)g^{-x_d} = 1$. Since $j^*, x_d$ are outside of $\hat{\mathsf{R}}$'s view and our base assumption is that $\hat{\mathsf{R}}$ that makes at least one request on $\sigma_i \in [1, N]$, the probability that $\mathcal{B}$ does not abort is $\geq \frac{p-1}{p} \cdot \frac{1}{N}$. Thus, if no p.p.t. algorithm solves 3DDH with probability $> \epsilon$, then Adv [Game 2 ]- Adv [Game 1 ] $\leq \frac{Np\epsilon}{p-1}$. □

**Lemma 3.** *If no p.p.t adversary has advantage $> \epsilon$ at solving the 3DDH problem, then*

$$Adv\ [Game\ 3] - Adv\ [Game\ 2] \leq N\epsilon.$$

*Proof.* See the proof of Lemma A.3 in [18]. □

Receiver Security. For any real-world cheating sender $\hat{\mathsf{S}}$ we can construct an ideal-world sender $\hat{\mathsf{S}}'$ such that all p.p.t. distinguishers have negligible advantage at distinguishing the distribution of the real and ideal experiments. Let us now describe the operation of $\hat{\mathsf{S}}'$, which runs $\hat{\mathsf{S}}$ internally, interacting with it in the role of the Receiver.

1) To begin, $\hat{\mathsf{S}}'$ runs the $\mathsf{R_I}$ algorithm, with the following modification: when $\hat{\mathsf{S}}$ proves knowledge of $a$, $\hat{\mathsf{S}}'$ uses the knowledge extractor $\mathsf{E}_1$ to extract $a$, outputting error if the extractor fails. Otherwise, it has obtained the values $(pk, C_1, \cdots, C_N)$.

2) For $i = 1$ to $N$, $\hat{\mathsf{S}}'$ decrypts each of $\hat{\mathsf{S}}$'s ciphertexts $C_1, \cdots, C_N$ using the value a as a decryption key, and sends the resulting $M_1^*, \cdots, M_N^*$ to the trusted party $\mathsf{T}$.

3) Whenever $\mathsf{T}$ indicates to $\hat{\mathsf{S}}'$ that a transfer has been initiated, $\hat{\mathsf{S}}'$ runs the transfer protocol with $\hat{\mathsf{S}}$ on the fixed index 1. If the transfer succeeds, $\hat{\mathsf{S}}'$ returns the bit 1 (success) to $\mathsf{T}$, or 0 otherwise.

4) $\hat{\mathsf{S}}'$ uses $\hat{\mathsf{S}}$'s output as its own.

**Theorem 2.** *Let $\epsilon_{WI}$ be the maximum advantage that any p.p.t. algorithm has at distinguishing a WIPoK, and let $\epsilon_{Ext}$ be the maximum probability that the extractor $\mathsf{E}_1$ fails. Then $\forall$ p.p.t. D:*

$$Pr[D(Real_{\hat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \cdots, M_N, \Sigma)) = 1] -$$

$$Pr[D(Ideal_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \cdots, M_N, \Sigma)) = 1]$$

$$\leq (k+1)\epsilon_{Ext} + k\epsilon_{WI}.$$

*Proof.* Refer to the proof of Theorem 3.3 in [18]. □

## 5 Conclusion

In this paper, we present a review on the Green-Hohenberger adaptive OT scheme and put forth a concrete improvement which is based on 3DDH assumption in bilinear groups. We show that in the original scheme there are some redundancies. Using the modified simulation which needs more less parameters than the simulation presented in the original paper, we prove that the improvement keeps secure under 3DDH assumption. This is a more simple assumption than $q$-power DDH assumption and $q$-strong DH assumption for [6], Decision Linear $q$-Hidden LRSW assumption for [17], Decisional $n$th Residuosity assumption for [23], Comp. Dec. Residuosity assumption and $q$-DDHI assumption for [21], DLIN assumption, $q$-Hidden SDH assumption and $q$-TDH assumption for [34]. The skills developed in the paper, we believe, is helpful for optimizing other cryptographic schemes.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project 61303200, 61411146001), the Shanghai Leading Academic Discipline Project (S30104). The authors gratefully acknowledge the reviewers for their valuable suggestions.

## References

[1] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," in *Proceedings of Advances in Cryptology (CRYPTO'89)*, pp. 547–557, Santa Barbara, USA, Aug. 1989.

[2] M. K. Bhatia1, S. K. Muttoo, and M. P.Bhatia, "Secure requirement prioritized grid scheduling model," *International Journal of Network Security*, vol. 15, no. 6, pp. 478–483, 2013.

[3] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proceedings of Advances in Cryptology (EUROCRYPT'04)*, pp. 56–73, Interlaken, Switzerland, May 2004.

[4] D. Boneh, A. Sahai, and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *Proceedings of Advances in Cryptology (EUROCRYPT'06)*, pp. 573–592, St. Petersburg, Russia, May 2006.

[5] G. Brassard, C. Crepeau, and J. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of Advances in Cryptology (CRYPTO'86)*, pp. 234–238, Santa Barbara, USA, Aug. 1986.

[6] J. Camenisch, G. Neven, and A. Shelat, "Simulatable adaptive oblivious transfer," in *Proceedings of Advances in Cryptology (EUROCRYPT'07)*, pp. 573–590, Barcelona, Spain, May 2007.

[7] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Proceedings of Advances in Cryptology (CRYPTO'97)*, pp. 410–424, Santa Barbara, California, Aug. 1997.

[8] Z. J. Cao and H. Y. Cao, "Improvement of Camenisch-Neven-Shelat oblivious transfer scheme," *International Journal of Network Security*, vol. 17, no. 2, pp. 103–109, 2015.

[9] Z. J. Cao, F. Lafitte, and O. Markowitch, "A note on a signature building block and relevant security reduction in the Green-Hohenberger OT scheme," in *Proceedings of 9th International Conference on-Information Security and Cryptology (Inscrypt'13)*, pp. 282–288, Guangzhou, China, Nov. 2013.

[10] C. C. Chang and Y. P. Lai, "Efficient t-out-of-n oblivious transfer schemes," in *Proceedings of the 2008 International Conference on Security Technology*, pp. 3–6, Hainan, China, Dec. 2008.

[11] C. C. Chang and J. S. Lee, "Robust t-out-of-n oblivious transfer mechanism based on CRT," *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 226–235, 2009.

[12] C. K. Chu and W. G. Tzeng, "Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries," in *Proceedings of 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05)*, pp. 172–183, Les Diablerets, Switzerland, Jan. 2005.

[13] C. K. Chu and W. G. Tzeng, "Efficient k-out-of-n oblivious transfer schemes," *Journal of Universal Computer Science*, vol. 14, no. 3, pp. 397–415, 2008.

[14] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[15] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," in *Proceedings of 19th Annual ACM Conference on Theory of Computing (STOC'87)*, pp. 218–229, New York,USA, May 1987.

[16] M. Green and S. Hohenberger, "Blind identity-based encryption and simulatable oblivious transfer," in *Proceedings of Advances in Cryptology (ASIACRYPT'07)*, pp. 265–282, Kuching, Malaysia, Dec. 2007.

[17] M. Green and S. Hohenberger, "Universally composable adaptive oblivious transfer," in *Proceedings of Advances in Cryptology (ASIACRYPT'08)*, pp. 179–197, Melbourne, Australia, Dec. 2008.

[18] M. Green and S. Hohenberger, "Practical adaptive oblivious transfer from simple assumptions," in *Proceedings of the Eighth Theory of Cryptography Conference (TCC'11)*, pp. 347–363, Brown University,USA, Mar. 2011.

[19] S. Hohenberger and B. Waters, "Realizing hash-and-sign signatures under standard assumptions," in *Proceedings of Advances in Cryptology (EUROCRYPT'09)*, pp. 333–350, Cologne, Germany, Apr. 2009.

[20] A. Jain and C. Har, "A new efficient protocol for k-out-of-n oblivious transfer," *Cryptologia*, vol. 34, no. 4, pp. 282–290, 2010.

[21] S. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection," in *Proceedings of the Sixth Theory of Cryptography Conference (TCC'09)*, pp. 577–594, San Francisco, USA, Mar. 2009.

[22] M. Kumar, M. K. Gupta, and S. Kumari, "An improved efficient remote password authentication scheme with smart card over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167–177, 2011.

[23] K. Kurosawa and R. Nojima, "Simple adaptive oblivious transfer without random oracle," in *Proceedings of Advances in Cryptology (ASIACRYPT'09)*, pp. 334–346, Tokyo, Japan, Dec. 2009.

[24] K. Kurosawa, R. Nojima, and T. P. Le, "Efficiency-improved fully simulatable adaptive OT under the DDH assumption," in *Proceedings of 7th Conference on Security and Cryptography for Networks (SCN'10)*, pp. 172–181, Amalfi, Italy, Sep. 2010.

[25] K. Kurosawa, R. Nojima, and T. P. Le, "Generic fully simulatable adaptive oblivious transfer," in *Proceedings of 9th International Conference on Applied Cryptography and Network Security (ACNS'11)*, pp. 274–291, Nerja, Spain, June 2011.

[26] H. Lipmaa, "An oblivious transfer protocol with log-squared communication," in *Proceedings of 8th International Conference on Information Security (ISC'05)*, pp. 314–328, Singapore, Sep. 2005.

[27] Y. J. Liu, C. C. Chang, and S. C. Chang, "An efficient oblivious transfer protocol using residue number system," *International Journal of Network Security*, vol. 15, no. 3, pp. 212–218, 2013.

[28] G. Manikandan, M. Kamarasan, and N. Sairam, "A new approach for secure data transfer based on wavelet transform," *International Journal of Network Security*, vol. 15, no. 2, pp. 106–112, 2013.

[29] Y. Mu, J. Q. Zhang, and V. Varadharajan, "m out of n oblivious transfer," in *Proceedings of Information Security and Privacy, 7th Australian Conference (ACISP'02)*, pp. 395–405, Melbourne, Australia, July 2002.

[30] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," in *Proceedings of 31th Annual ACM Conference on Theory of Computing (STOC'99)*, pp. 245–254, Atlanta, Georgia, USA, May 1999.

[31] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proceedings of Advances in Cryptology (CRYPTO'99)*, pp. 573–590, Santa Barbara,USA, Aug. 1999.

[32] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pp. 448–457, Washington, D.C., USA, Jan. 2001.

[33] M. Rabin, "How to exchange secrets by oblivious transfer," Technical Report TR-81, May 1981.

[34] A. Rial, M. Kohlweiss, and B. Preneel, "Universally composable adaptive priced oblivious transfer," in *Proceedings of the Third International Conference on Pairing-based Cryptography (Pairing'09)*, pp. 231–247, Palo Alto, CA, USA, Aug. 2009.

[35] W. G. Tzeng, "Efficient 1-out-of-n oblivious transfer protocols with universally usable parameter," *IEEE Transactions on Computers*, vol. 53, no. 2, pp. 232–240, 2004.

[36] Q. Wu, J. H. Zhang, and Y. M. Wang, "Practical t-out-n oblivious transfer and its applications," *Information and Communications Security*, vol. 2836, pp. 226–237, 2003.

[37] A. Yao, "How to generate and exchange secrets," in *Proceedings of 27th Annual Symposium on Foundations of Computer Science (FOCS'86)*, pp. 162–167, Toronto, Canada, Oct. 1986.

[38] B. Zeng and et al., "A practical framework for t-out-of-n oblivious transfer with security against covert adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 465–479, 2012.

[39] B. S. Zhang, "Simulatable adaptive oblivious transfer with statistical receiver's privacy," in *Proceedings of the 5th International Conference on Provable Security (ProvSec'11)*, pp. 52–67, Xi'an, China, Oct. 2011.

**Zhengjun Cao** is an associate professor of department of Mathematics at Shanghai University. He received his Ph.D. degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He served as a post-doctor in Department of Computer Science, Universit Libre de Bruxelles, from 2008 to 2010. His current research interests include cryptography, discrete logarithms and quantum computation.

**Lihua Liu** is an associate professor of department of Mathematics at Shanghai Maritime University. She received her Ph.D. degree in applied mathematics from Shanghai Jiao Tong University. She was a visiting scholar in Department of Computer Science, Lakehead University, Canada, from 2013 to 2014. Her current research interests include combinatorics, cryptography and information security.

# An Efficient Batch Verifying Scheme for Detecting Illegal Signatures

Yanli Ren[1], Shuozhong Wang[1], Xinpeng Zhang[1], Min-Shiang Hwang[2,3]

*(Corresponding author: Min-Shiang Hwang)*

School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China[1]

Department of Computer Science and Information Engineering, Asia University[2]

No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan

(Email: mshwang@asia.edu.tw)

Department of Medical Research, China Medical University Hospital, China Medical University[3]

No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan

*(Received Dec. 25, 2012; revised and accepted May 14 & July 20, 2014)*

## Abstract

In a batch verifying scheme, multiple RSA digital signatures can be verified simultaneously in just one exponential operation time. Currently, the verifier could not easily detect where the signature-verification fault was located in most schemes, if the batch verification fails. In this article, we proposed a new batch verifying multiple RSA digital signatures scheme based on a cube. The scheme can detect accurately where the signature-verification fault is located. Moreover, once the total number of signatures is fixed, the size of exponentiation operations is independent from the number of illegal signatures in our scheme. Therefore, our scheme has a better performance and higher efficiency than the previous ones. Finally, we described an extend batch verifying scheme under the condition of $n$-dimension.

*Keywords: Batch verifying, digital signature, multiple signatures*

## 1 Introduction

RSA is a well-known public key cryptosystem where each user has a public key $e$ for encryption (verification) and a private key $d$ for decryption (signature) [17,18]. It protects the transaction information over the network [16], and satisfies the requirement of user authentication and communication security on networking environments [15]. In a RSA signature scheme, the signer uses the personal private key $d$ to sign document $M$ and obtains the signature $S = M^d$, and then the receiver verifies whether $M = S^e$ using the signer's public key $e$. If there are $t$ documents and signatures $(M_i, S_i)(i = 1, \cdots, t)$, the receiver then needs to verify these signatures one by one and fully executes $t$ exponential computations [11]. This will reduce the computer host's processing ability and the efficiency of RSA signature scheme. Therefore, the concept of batch verifying signatures has been introduced to efficiently improve the performance of verifying multiple RSA signatures [2,6,8,13,19,20].

Harn proposed a batch verifying multiple RSA signature scheme [5] in 1998, where multiple signatures could be verified simultaneously in just one exponential operation time. Such method is considered to be more efficient than the previous signature schemes where the signer must repeatedly verify each signature [3,12]. However, Hwang et al. showed that the Harn's scheme could not resist two kinds of attacks [7,10]. In addition, the verifier could not detect where the signature-verification fault was located if the batch verification fails in Harn's scheme. Since the verifier must re-verify each of the signatures and then confirms where the signature-verification fault is located, it is inefficient to detect the illegal signatures. There are many batch verifying multiple RSA signature schemes have been proposed [1,4,12,21].

Recently, Li et al. proposed a matrix-based solution to quickly find out where the signature-verification faults are located without re-verifying each of the signatures [14]. In their scheme, the performance would be at its best when both numbers of row and column square roots are equal to the message's numbers. Let's assume there are 25 signatures, the scheme is the most efficient one and the verifier needs to execute 10 exponential computations when the matrix has 5 rows and 5 columns. If there is one illegal signature, the verifier needs to execute 10 exponential computations. If there are two illegal signatures, the verifier needs to execute 14 exponential computations to detect the illegal signatures.

In this paper, we present a new batch verifying scheme which is especially efficient when there are illegal signatures. When the verifier receives $t$ signatures, it generates a cube of side length $n$ and fills these $t$ signatures in the

$n \times n \times n$ cube, where $n$ is the smallest integer which satisfies $n^3 \geq t$. Let's assume there are 25 signatures, the verifier generates a cube of side length 3 and executes $3 + 3 + 3 = 9$ exponential computations since 3 is the smallest integer which satisfies $3^3 \geq 25$. Moreover, the verification time would not increase as the number of the illegal signatures increases.

The paper is organized as follows: In Section 2, we review two batch verifying schemes including Harn's and Li's scheme. Then we present the proposed scheme and compare its performance with that of previous schemes in Sections 3 and 4, respectively. In Section 5, an extended batch verification scheme is described. Finally, we conclude our paper in Section 6.

# 2 Two Batch Verifying Scheme

We review two batch verifying schemes before presenting the proposed one.

## 2.1 Harn's Scheme

In this section, we first introduce Harn's batch verifying scheme [5]. Let's assume $p$ and $q$ are two prime numbers, and $N = pq$. $e$ and $d$ are presented as the signer's public key and private key respectively, which satisfies $ed \equiv 1 \mod \varphi(N)$, and $\varphi(\cdot)$ is the Euler function. $h(\cdot)$ is a public one-way hash function.

We suppose Alice sends the messages $M_0$, $M_1$, $\cdots$, $M_{t-1}$ and signatures $S_0$, $S_1$, $\cdots$, $S_{t-1}$ to Bob, where $S_i = h(M_i)^d \mod N$, $(i = 0, 1, \cdots, t-1)$. Bob can verify these signatures using Alice's public key $e$ by the following equation:

$$(\prod_{i=0}^{t-1} S_i)^e \overset{?}{=} \prod_{i=0}^{t-1} h(M_i). \tag{1}$$

If Equation (1) holds, $(S_0, S_1, \cdots, S_{t-1})$ are valid signatures of $M_0$, $M_1$, $\cdots$, $M_{t-1}$, respectively. In Harn's scheme, these signatures can be verified simultaneously in one exponential operation time.

## 2.2 Li et al.'s Scheme

The scheme was proposed by Li et al. recently [14]. When the verifier receives the messages $(M_1, S_1)$, $(M_2, S_2)$, $\cdots$, $(M_t, S_t)$ from the signer, the verifier will generate an $m \times n$ matrix (where $m \times n \geq t$) and $t$ random numbers $r_i$, $i = 1, 2, \ldots, t$, where $r_i \in \{1, 2, \ldots, t\}$. He then randomly fills these $t$ messages into the $m \times n$ matrix using the following equation (see Table 1):

$$S(m,n) = \begin{cases} S(\lceil r_i/n \rceil, n), & \text{if } r_i \bmod n = 0 \\ S(\lceil r_i/n \rceil, r_i \bmod n), & \text{otherwise.} \end{cases} \tag{2}$$

Table 1: An $m \times n$ matrix

| | | | | |
|---|---|---|---|---|
| S(1,1) | S(1,2) | ... | S(1,n-1) | S(1,n) |
| S(2,1) | S(2,2) | ... | S(2,n-1) | S(2,n) |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| S(m-1,1) | S(m-1,2) | ... | S(m-1,n-1) | S(m-1,n) |
| S(m,1) | S(m,2) | ... | S(m,n-1) | S(m,n) |

After filling these messages in the $m \times n$ matrix, the verifier could batch verify each of the rows and the columns, respectively. The complete batch verifying process is divided into two verifications: row verification and column verification. The details of row and column verification are shown as follows.

- Row verification:

$$\begin{gathered} \text{First row: } (\prod_{i=1}^{n} S_{(1,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(1,i)}) \bmod N, \\ \text{Second row: } (\prod_{i=1}^{n} S_{(2,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(2,i)}) \bmod N, \\ \vdots \\ m\text{-th row: } (\prod_{i=1}^{n} S_{(m,i)})^e \overset{?}{=} \prod_{i=1}^{n} h(M_{(m,i)}) \bmod N. \end{gathered}$$

- Column verification:

$$\begin{gathered} \text{First column: } (\prod_{i=1}^{m} S_{(i,1)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,1)}) \bmod N, \\ \text{Second column: } (\prod_{i=1}^{m} S_{(i,2)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,2)}) \bmod N, \\ \vdots \\ n\text{-th column: } (\prod_{i=1}^{m} S_{(i,n)})^e \overset{?}{=} \prod_{i=1}^{m} h(M_{(i,n)}) \bmod N. \end{gathered}$$

If there are some signature-verification faults in the matrix, we could find out where these signature-verification faults are located by finding the matrix positions of row and column overlaps.

Table 2: An $5 \times 5$ matrix

| | | | | |
|---|---|---|---|---|
| S(1,1) | S(1,2) | S(1,3) | S(1,4) | S(1,5) |
| S(2,1) | S(2,2) | S(2,3) | S(2,4) | S(2,5) |
| S(3,1) | S(3,2) | S(3,3) | S(3,4) | S(3,5) |
| S(4,1) | S(4,2) | S(4,3) | S(4,4) | S(4,5) |
| S(5,1) | S(5,2) | S(5,3) | S(5,4) | S(5,5) |

Suppose Alice sends 25 messages to Bob, then Bob will generate 25 random numbers and a $5 \times 5$ matrix shown as Table 2. After batch verifying each of the rows and the columns, Bob could easily realize there was a signature-verification fault occurring and precisely detects where the signature-verification fault is located. Assume there was one signature-verification fault in the position $S(3,3)$ of matrix, there would occur two verification fails and these two fails would occur in the third row and the third column, respectively. According to the verification fails of the third row and the third column overlaps, the signature-verification fault could be precisely detected in the position $S(3,3)$ of matrix. However, it is possible for the verifier to execute additional operations if two illegal signatures occur. As shown in [14], the number of total verification is 10 if two signature-verification faults are occurring on the same row or on the same column, and the number is 14 if two illegal signatures are occurring on adjacent diagonal or not occurring on the same row or not on the same column. Please refer to [14] for more details.

## 3 The Proposed Scheme

We now present a batch verifying multiple signatures scheme which is more efficient than the previous ones, especially when the illegal signature occurs. The details of our scheme are described as follows.

First, the verifier generates a cube with side length $m$ when he receives some pairs of message and signature $(M_0, S_0), (M_1, S_1), \cdots, (M_{t-1}, S_{t-1})$ from the signer, where $m$ is the smallest integer which satisfies $m^3 \geq t$.

Next, the verifier chooses $t$ random numbers $r_i$, where $r_i \in \{0, 1, \cdots, m^3 - 1\}$, $i = 0, 1, \cdots, t-1$, and fills these $t$ signatures in the $m \times m \times m$ cube according to coordinate figure $(x, y, z)$, where

$$r_i = xm^2 + ym + z, \quad \text{and} \quad x, y, z \in \{0, 1, \cdots, m-1\}. \quad (3)$$

Finally, the verifier could then batch verify each plane according to the three coordinate axes. The details are shown as follows.

- $x$-axis plane:

$$x = 0:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(0,i,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(0,i,j)}),$$
$$x = 1:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(1,i,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(1,i,j)}),$$
$$\vdots$$
$$x = m-1:$$
$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(m-1,i,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(m-1,i,j)}).$$

- $y$-axis plane:

$$y = 0:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,0,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,0,j)}),$$
$$y = 1:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,1,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,1,j)}),$$
$$\vdots$$
$$y = m-1:$$
$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,m-1,j)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,m-1,j)}).$$

- $z$-axis plane:

$$z = 0:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,j,0)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,j,0)}),$$
$$z = 1:\ (\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,j,1)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,j,1)}),$$
$$\vdots$$
$$z = m-1:$$
$$(\prod_{i=0}^{m-1}\prod_{j=0}^{m-1} S_{(i,j,m-1)})^e \overset{?}{=} \prod_{i=0}^{m-1}\prod_{j=0}^{m-1} h(M_{(i,j,m-1)}).$$

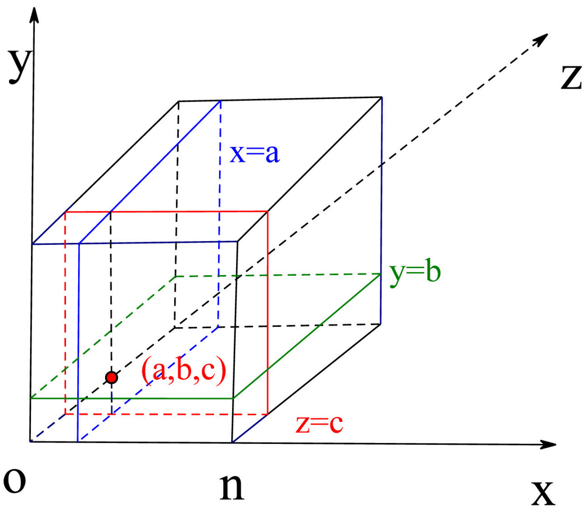


Figure 1: An $m \times m \times m$ cube

If there are some signature-verification faults in the cube, we could find out where these faults are located by finding the point of intersection of three kinds of plane. As shown in Figure 1, there is a signature-verification fault in the position $(a, b, c)$ of the cube if three verifications fail in the $x = a$, $y = b$, and $z = c$ plane, respectively.

We will now give a simple example to show the correctness of our scheme. Let's suppose Alice have sent 64 messages to Bob, then Bob will choose 64 random numbers and generate a $4 \times 4 \times 4$ cube as shown in Figure 2. If $r_0 = 22$, the pair $(M_0, S_0)$ would be filling in the position $(1, 1, 2)$ of the cube since $22 = 1 \cdot 4^2 + 1 \cdot 4 + 2$. If $r_1 = 45$, the pair $(M_1, S_1)$ would be filling in the position $(2, 3, 1)$ of the cube because $45 = 2 \cdot 4^2 + 3 \cdot 4 + 1$. The rest can be deduced similarly by Equation (2). After filling 64 signatures in the cube, Bob could then batch verify three kinds of plane by the method described above.

- $x$-axis plane:

$$x = 0:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(0,i,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(0,i,j)}),$$
$$x = 1:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(1,i,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(1,i,j)}),$$
$$x = 2:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(2,i,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(2,i,j)}),$$
$$x = 3:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(3,i,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(3,i,j)}).$$

- $y$-axis plane:

$$y = 0:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,0,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,0,j)}),$$
$$y = 1:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,1,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,1,j)}),$$
$$y = 2:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,2,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,2,j)}),$$
$$y = 3:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,3,j)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,3,j)}).$$

- $z$-axis plane:

$$z = 0:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,j,0)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,j,0)}),$$
$$z = 1:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,j,1)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,j,1)}),$$
$$z = 2:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,j,2)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,j,2)}),$$
$$z = 3:\ (\prod_{i=0}^{3}\prod_{j=0}^{3} S_{(i,j,3)})^e \overset{?}{=} \prod_{i=0}^{3}\prod_{j=0}^{3} h(M_{(i,j,3)}).$$

Table 3: Experiment results of RSA, Harn, Li, and our schemes

| Method | Message | Size of Row | Size of $x$ | Size of |
|---|---|---|---|---|
| | | (Column) | $(y, z)$-axis | exponentiation |
| RSA [18] | 25 | — | — | 25 |
| Harn [5] | 25 | — | — | 1 |
| Li [14] | 25 | 5 | — | 10 |
| Ours | 25 | — | 3 | 9 |
| Method | Message | Size of Row | Size of $x$ | Size of |
| | | (Column) | $(y, z)$-axis | exponentiation |
| RSA [18] | 100 | — | — | 100 |
| Harn [5] | 100 | — | — | 1 |
| Li [14] | 100 | 10 | — | 20 |
| Ours | 100 | — | 5 | 15 |
| Method | Message | Size of Row | Size of $x$ | Size of |
| | | (Column) | $(y, z)$-axis | exponentiation |
| RSA [18] | 256 | — | — | 256 |
| Harn [5] | 256 | — | — | 1 |
| Li [14] | 256 | 16 | — | 32 |
| Ours | 256 | — | 7 | 21 |

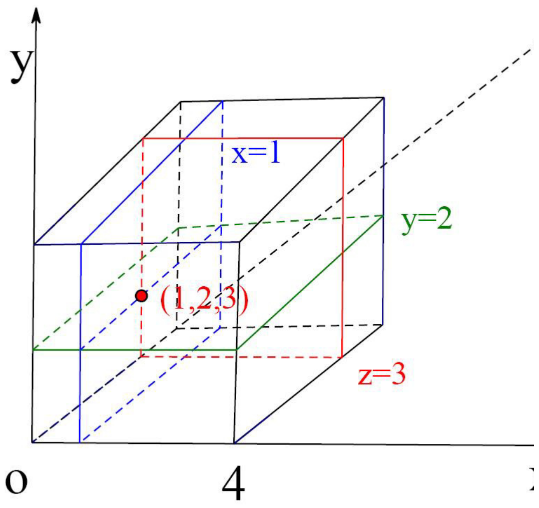


Figure 2: An $4 \times 4 \times 4$ cube

After batch verifying each plane, Bob is now confirmed whether the signature-verification fault is occurring or not. We suppose there was one signature-verification fault in the position $(1, 2, 3)$ of the cube, so Bob could realize there was a signature-verification fault occurring and precisely detects where the signature-verification fault is located. From the method described above, there would occur three verification that fails in the $x = 1$, $y = 2$, and $z = 3$ plane, respectively. According to the point of intersection of three kinds of plane, the signature-verification fault could be precisely detected in the position $(1, 2, 3)$ of the cube.

# 4 Implementation and Result Analysis

## 4.1 Experimental Results

In this section, we implement RSA, Harn's scheme, Li's scheme, and ours, with the experimental results of four schemes shown in Table 3. From Table 3, we have concluded that Harn's scheme is better when the batch verification of multiple signatures have succeeded. However, it must re-verify all signatures if there are illegal ones as presented in Section 1. Thus, the performance of Harn's scheme is worse than Li's and ours when illegal signatures occur. In addition, our scheme needs less exponentiation operations for the same number of messages. Therefore, the performance of our scheme is best in most situations.

In Table 3, we compared the sizes of exponentiation operations for different number of signatures in all four schemes. In RSA scheme, the verifier needs to verify the signatures one by one, so the sizes of exponentiation operations are 25, 100, and 256, respectively. As described in Section 2, the verifier can determine the correctness of signature by using one exponential operation in Harn's scheme. Therefore, we only need to show the size of exponentiation operations in Li's and our scheme. In Li's scheme, the verifier can obtain $5 \times 5, 10 \times 10$, and $16 \times 16$ matrixes and executes $5 + 5 = 10, 10 + 10 = 20$, and $16 + 16 = 32$ exponentiation operations for $25, 100$, and 256 signatures since $5, 10$ and 16 are the square root of 25, 100, and 256, respectively. In our scheme, the verifier can generate $3 \times 3 \times 3, 5 \times 5 \times 5$, and $7 \times 7 \times 7$ cubes and executes $3 + 3 + 3 = 9$, $5 + 5 + 5 = 15$, and $7 + 7 + 7 = 21$ exponentiation operations since 3, 5, 7 are the smallest integer which satisfies $3^3 \geq 25$, $5^3 \geq 100$, and $7^3 \geq 256$,

Table 4: Comparisons for detecting illegal signatures among RSA, Harn, Li, and our schemes

| Method | Message | Size of exponentiation (one illegal signature) | Size of exponentiation (two illegal signatures) |
|---|---|---|---|
| RSA [18] | 25 | 25 | 25 |
| Harn [5] | 25 | 26 | 26 |
| Li [14] | 25 | 10 | 14 |
| Ours | 25 | 9 | 9 |

| Method | Message | Size of exponentiation (one illegal signature) | Size of exponentiation (two illegal signatures) |
|---|---|---|---|
| RSA [18] | 100 | 100 | 100 |
| Harn [5] | 100 | 101 | 101 |
| Li [14] | 100 | 20 | 24 |
| Ours | 100 | 15 | 15 |

| Method | Message | Size of exponentiation (one illegal signature) | Size of exponentiation (two illegal signatures) |
|---|---|---|---|
| RSA [18] | 256 | 256 | 256 |
| Harn [5] | 256 | 257 | 257 |
| Li [14] | 256 | 32 | 36 |
| Ours | 256 | 21 | 21 |

respectively.

### 4.2 Analysis of Illegal Signature Detection

In this section, we now present the size of exponentiation operations in four schemes when illegal signatures occur. As described in Section 1, the performance of illegal signatures detection is regarded as the position it is located in. From Table 4, we know that our scheme is better than Harn's and Li's for determining the situation where illegal signatures are located. Once the total number of signatures is fixed, the size of exponentiation operations is independent from the number of illegal signatures in our scheme.

In Table 4, we compared the sizes of exponentiation for detecting one and two illegal signatures among RSA, Harn, Li, and our schemes. In RSA scheme, the verifier needs to verify the signatures one by one, so the sizes of exponentiation operations are 25, 100, and 256 respectively. In Harn's scheme, the verifier must re-verify each signatures if there are illegal ones, so he needs to execute 26, 101, and 257 exponentiation operations, respectively. As shown in Section 2.2, in Li's scheme, one illegal signature can be detected accurately after batch verification finished, and the verifier must add 4 exponentiation operations if there are two illegal signatures. From Table 3, we know that 10, 20, and 32 operations are needed for 25, 100, and 256 signatures respectively in Li's scheme. Thus, the verifier executes $10, 20$, and 32 operations when one illegal signature occurs; and $14, 24$, and 36 operations if there are two illegal signatures in Li's scheme. In our scheme, the position of illegal ones can be determined accurately once batch verification is finished and the size of operations are independent from the number of signatures. From Table 3, we know that $9, 15$, and 21 operations are needed for $25, 100$, and 256 signatures respectively in our scheme. Therefore, the sizes of exponentiation operations are $9, 15$, and 21 in our scheme whether one or two faults occurred.

## 5 The Extension of the Scheme

The proposed scheme is based on a cube, and we can extend it to the condition of $n$-dimension. First, the verifier generates an $n$-dimension object with side length $m$ when he receives some pairs of message and signature $(M_0, S_0), (M_1, S_1), \cdots, (M_{t-1}, S_{t-1})$ from the signer, where $m$ is the smallest integer which satisfies $m^n \geq t$.

Next, the verifier chooses $t$ random numbers $r_i$, where $r_i \in \{0, 1, \cdots, m^n - 1\}$, $i = 0, 1, \cdots, t-1$, and fills these $t$ messages in the $m^n$ object according to coordinate figure $(a_{n-1}, a_{n-2}, \cdots, a_1, a_0)$, where $a_{n-1}, \cdots, a_1, a_0 \in \{0, 1, \cdots, m-1\}$ and

$$r_i = a_{n-1}m^{n-1} + a_{n-2}m^{n-2} + \cdots + a_1 m + a_0.$$

Finally, the verifier could then batch verify each plane according to $n$-dimension coordinate axis. The details are described as follows.

1) $a_{n-1}$-axis plane:

   a. $a_{n-1} = 0$:

$$\Big(\prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(0,a_{n-2},\cdots,a_0)}\Big)^e \overset{?}{=} \prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(0,a_{n-2},\cdots,a_0)}).$$

b. $a_{n-1} = 1$:

$$(\prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(1,a_{n-2},\cdots,a_0)})^e \overset{?}{=} \prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(1,a_{n-2},\cdots,a_0)}).$$

$$\vdots \qquad\qquad \vdots$$

c. $a_{n-1} = m-1$:

$$(\prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(m-1,a_{n-2},\cdots,a_0)})^e \overset{?}{=} \prod_{a_{n-2}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(m-1,a_{n-2},\cdots,a_0)}).$$

2) $a_{n-2}$-axis plane:

a. $a_{n-2} = 0$:

$$(\prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(a_{n-1},0,\cdots,a_0)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(a_{n-1},0,\cdots,a_0)}).$$

b. $a_{n-2} = 1$:

$$(\prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(a_{n-1},1,\cdots,a_0)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(a_{n-1},1,\cdots,a_0)})$$

$$\vdots \qquad\qquad \vdots$$

c. $a_{n-2} = m-1$:

$$(\prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} S_{(a_{n-1},m-1,\cdots,a_0)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \prod_{a_{n-3}=0}^{m-1} \cdots \prod_{a_0=0}^{m-1} h(M_{(a_{n-1},m-1,\cdots,a_0)}).$$

$$\vdots$$

3) $a_0$-axis plane:

a. $a_0 = 0$:

$$(\prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} S_{(a_{n-1},\cdots,a_1,0)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} h(M_{(a_{n-1},\cdots,a_1,0)}).$$

b. $a_0 = 1$:

$$(\prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} S_{(a_{n-1},\cdots,a_1,1)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} h(M_{(a_{n-1},\cdots,a_1,1)}).$$

$$\vdots \qquad\qquad \vdots$$

c. $a_0 = m-1$:

$$(\prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} S_{(a_{n-1},\cdots,a_1,m-1)})^e \overset{?}{=} \prod_{a_{n-1}=0}^{m-1} \cdots \prod_{a_1=0}^{m-1} h(M_{(a_{n-1},\cdots,a_1,m-1)}).$$

Therefore, the total number of exponentiation operations is $mn$ in the extended batch verification scheme. If there are some signature-verification faults in the $n$-dimension object, we could find out where these faults are located by finding the point of intersection of $n$ kinds of plane. For example, there is a signature-verification fault in the position $(0, 1, \cdots, m-1)$ of the $n$-dimension object, if $n$ verifications failed in the $a_{n-1} = 0$ plane, $a_{n-2} = 1$ plane, $\cdots$ and $a_0 = m-1$ plane, respectively.

## 6 Conclusions

We presented a new batch verification multiple RSA signatures scheme which fills the signatures into a cube. It can detect accurately where the illegal signatures are located without additional re-verify operations. Moreover, the verification time would not increase as the number of the illegal signatures increases in one batch verification. Experiment shows our scheme is more efficient than the previous schemes, especially when the number of the signatures is very large. We then extended this scheme to the condition of $n$-dimension.

## Acknowledgements

The work described in this paper was supported by the Natural Science Foundation of China (Grant No. 61202367, 61472235), Natural Science Foundation of Shanghai (Grant No. 12ZR1443700), and the Innovation Program of Shanghai Municipal Education Commission (Grant No. 14YZ020).

## References

[1] F. Bao, C. C. Lee, M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple RSA digital signatures," *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 1195-1200, 2006.

[2] T. Cao, D. Lin, and R. Xue, "Security analysis of some batch verifying signatures from pairings," *International Journal of Network Security*, vol. 3, no. 2, pp. 138-143, 2006.

[3] S. W. Changchien, M. S. Hwang, "A batch verifying and detecting multiple RSA digital signatures," *International Journal of Computational and Numerical Analysis and Applications*, vol. 2, no. 3, pp. 303-307, Oct. 2002.

[4] T. Y. Chang, M. S. Hwang, W. P. Yang, and K. C. Tsou, "A modified Ohta-Okamoto digital signature for batch verification and its multi-signature version," *International Journal of Engineering and Industries*, vol. 3, no. 3, pp. 75-83, Sep. 2012.

[5] L. Harn, "Batch verifying multiple RSA digital signatures," *Electronics Letters*, vol. 34, no. 12, pp. 1219-1220, 1998.

[6] M. S. Hwang, K. F. Hwang, I. C. Lin, "Cryptanalysis of the batch verifying multiple RSA digital signatures," *Informatica*, vol. 11, no. 1, pp. 1-4, Jan. 2000.

[7] M. S. Hwang, I. C. Lin, and K. F. Hwang, "Cryptanalysis of the batch verifying multiple RSA digital signatures," *Informatica*, vol. 11, no. 1, pp. 15-19, 2000.

[8] M. S. Hwang and C. C. Lee, "Research issues and challenges for multiple digital signatures," *International Journal of Network Security*, vol. 1, no. 1, pp. 1-7, July 2005.

[9] M. S. Hwang, C. C. Lee, Y. C. Lai, "Traceability on RSA-based partially signature with low computation," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465-468, Dec. 2003.

[10] M. S. Hwang, C. C. Lee, E. J. L. Lu, "Cryptanalysis of the batch verifying multiple DSA-type digital signatures," *Pakistan Journal of Applied Sciences*, vol. 1, no. 3, pp. 287-288, July 2001.

[11] M. S. Hwang, E. J. L. Lu, I. C. Lin, "Practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1552-1560, Nov./Dec. 2003.

[12] S. J. Hwang, M. S. Hwang, and S. F. Tzeng, "A new digital multisignature scheme with distinguished signing authorities," *Journal of Information Science and Engineering*, vol. 19, no. 5, pp. 881-887, Sep. 2003.

[13] K. Kim, I. Yie, S. Lim, and D. Nyang, "Batch verification and finding invalid signatures in a group signature scheme," *International Journal of Network Security*, vol. 13, no. 2, pp. 61-70, 2011.

[14] C. T. Li, M. S. Hwang, "A batch verifying and detecting the illegal signatures," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 12, pp. 5311-5320, 2010.

[15] C. T. Li, M. S. Hwang, and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, 2009.

[16] C. T. Li, M. S. Hwang, and C. Y. Liu, "An electronic voting protocol with deniable authentication for mobile ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2534-2540, 2008.

[17] N. Ojha and S. Padhye, "Cryptanalysis of multi prime RSA with secret key greater than public key," *International Journal of Network Security*, vol. 16, no. 1, pp. 53-57, 2014.

[18] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[19] M. Stanek, "Attacking LCCC batch verification of RSA signatures," *International Journal of Network Security*, vol. 6, no. 2, pp. 238-240, 2008.

[20] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verification for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, pp. 77-84, 2011.

[21] J. Zhang, M. Xu, and L. Liu, "On the security of a secure batch verification with group testing for VANET," *International Journal of Network Security*, vol. 16, no. 4, pp. 313-320, 2014.

**Yanli Ren** is an associate professor in School of Communication and Information Engineering at Shanghai University, China. She was awarded a MS degree in applied mathematics in 2005 from Shaanxi Normal University, China, and a Ph.D. degree in computer science and technology in 2009 from Shanghai Jiao Tong University, China. Her research interests include applied cryptography, secure outsourcing computing, and network security.

**Shuozhong Wang** received BS degree in 1966 from Peking University, P.R. China, and Ph.D. degree in 1982 from University of Birmingham, England. He was with Institute of Acoustics, Chinese Academy of Sciences, from 1983 to 1985 as a research fellow. He joined Shanghai University of Technology in October 1985 as an associate professor. He is now a professor of the School of Communication and Information Engineering, Shanghai University. Professor Wang was a visiting associate scientist at Department of Electrical Engineering and Computer Science, University of Michigan, USA, from March 1993 to August 1994. His research interests include acoustics, image processing, audio processing, and information security.

**Xinpeng Zhang** received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He was with the State University of New York at Binghamton as a visiting scholar from January 2010 to January 2011, and Konstanz University as an experienced researcher sponsored by the Alexander von Humboldt Foundation from March 2011 to May 2012. His research interests include multimedia security, image processing, and digital forensics. He has published more than 170 papers in these areas.

**Min-Shiang Hwang** received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications.

# Fault-tolerant Verifiable Keyword Symmetric Searchable Encryption in Hybrid Cloud

Jie Wang, Xiao Yu, and Ming Zhao
*(Corresponding author: Ming Zhao)*

Aviation University of Air Force, 130022, Changchun, Jilin, China
(Email: michaelwangliu@163.com & zhaoming2014@mail.jlu.edu.cn)

*(Received Nov. 26, 2014; revised and accepted Feb. 10, & Mar. 16, 2015)*

## Abstract

As cloud computing is increasingly expanding its application scenario, it is vital for cloud storage customers not to sacrifice the confidentiality of sensitive data while making fullest use of operational functionality of cloud secure systems. Although traditional searchable encryption can well solve exact keyword search on encrypted data with retrieving files by search interest, it does not work when typos or misspelling mistakes occur. Many specific algorithms have been well proposed to solve this difficult problem. However, most of the schemes mainly focus on the single cloud to achieve fuzzy keyword search, which means that fuzzy-keyword index construction must take possible typos into account and makes existing exact-keyword index useless. In addition, existing searching schemes rarely take interaction between the data user and the cloud to improve system's usability and user's retrieval satisfactory degree into consideration. In this paper, we propose an improved scheme named as Distributed Fault-tolerant Keyword Search Supporting Verifiable Search-ability (DFKSSVS) in hybrid cloud with the emphasis of interaction circumstances. Through improved dictionary-based keyword construction scheme, we generate fuzzy keyword set, and build secure index for efficient fuzzy search. After searching procedures, the scheme can support verifiability of returned files via $proof$ returned by cloud as well, and interaction between data user and private cloud to achieve dynamic ranking of retrieval results statistically. Through rigorous security and thorough analysis, we show that the improved solution can meet verifiable fuzzy keyword search on cloud encrypted data with supporting the exact-keyword index already generated. Security analysis and extensive experimental results demonstrate the accuracy and efficiency of our proposed scheme.

*Keywords: Cloud storage, fault-tolerant keyword search, improved-dictionary-based fuzzy set, outsourcing data, searchable encryption, verifiable keyword search*

## 1 Introduction

Nowadays, the increasing growth of Big Data in IT industry impels the application expansion of cloud computing. As a typical application, cloud storage has gained popularity in many corporations and companies around the world. However, as the large amount of sensitive data, such as enterprise basic files, government investigation reports, private health records and so on, is in the out-of-control domain, data privacy has become the top concern of whether it is a must to outsource data to the cloud. Data encryption is an effective solution to keep its confidentiality, which has yet sacrificed data usability. Encryption can preserve outsourced data's confidentiality, integrity and accessibility (CIA) of cloud data, and no one can know the contents of encrypted files without decryption keys, however, secure cloud system usability is lowered for non-operation-ability on cipher-text.

The best solution for encrypted data computation is Fully Homomorphic Encryption (HHE), which allows users to operate directly on cipher-texts and then produce results of matching procedures. Gentry et al. [8] made a breakthrough in theoretical domain, but the scheme of construction efficiency is far from practical utilization. Moreover, data users are usually interested in the most relevant files whose ranking is in the top-k list rather than all files returned from cloud. From the perspective of information retrieval, users choose to input some specific keywords named as "keyword-based search" to selectively retrieve relevant files. Unfortunately, computable retrieval operations on encrypted data by keyword search are limited due to no suitable schemes on cipher-text search compared with traditional retrieval methods on plain-text search. Although encrypted keyword can protect its privacy, how to use plain-text search techniques on encrypted data turns to be a real problem, which attracts much more attention of researches on it. Different from the traditional Private Information Retrieval (PIR) schemes, an alternative, that is searchable encryption (SE) schemes, has been proposed and researched for a long time. SE is a key technique for data users to directly operate encrypted data, but traditional SE without secure index is inefficient facing with the large-scale cloud encrypted data. So, it is important to construct a secure index for encrypted data files.

Searchable encryption is an important and fundamental solution to solve the problems of encrypted data utilization, as well as integration of data confidentiality and usability. In general, searchable encryption can be divided into two subcategories, that is, Public Key Searchable Encryption (PKSE), and Symmetric Searchable Encryption (SSE). PKSE can support much more flexible search operations and complicated search applications, but more computationally huge overhead is produced because of many pairing-operations compared with SSE. In contrast, SSE depends on its computational efficiency and operational convenience to make it to be a research hot spot. No matter PKSE or SSE, keyword search is associated with index of files. By integrating trapdoors (encrypted form of searched keyword) with secure index (encrypted form of file index), effective keyword search can be finished while retrieval contents and search results are blind to cloud servers.

Furthermore, fuzzy keyword seems to be a hot topic in the plain-text information retrieval field, because retrievers may have typos by accident or statistical misspelling mistakes during retrieval procedures. As an applicable expansion, fuzzy keyword search on encrypted data has been researched actively. Li et al. [11] for the first time proposed wildcard-based keyword search over encrypted scheme, which is proven its weakness of insecurity by Zheng et al. in HPCC 2013 conference [20]. Wang et al. [18] suggested a solution using trie-tree for index construction, which has large space-cost of building index and infeasible updating of index tree. Chuah et al. [6] presented a scheme which has secure index by using bed-tree with low efficiency. Liu et al. [12] proposed a solution named as "dictionary-based fuzzy keyword search on encrypted data" with small index, but its fuzzy keyword set is not all-around, which means loss of many possible exact keywords to match with. Recently, Zhou et al. [21] proposed a different scheme to make fuzzy keyword set by utilizing k-gram. Wang et al. [16] aimed at multi-keyword fuzzy search on encrypted data by locality sensitive hashes and Bloom filters to support multi-keyword search with low search complexity. However, all the schemes face with retrieval efficiency problem and defective construction of fuzzy-keyword index which has made exact-keyword index already constructed useless.

Another issue to which needs to pay much attention is verifiability of returned encrypted data from the public cloud. This was first mentioned by Chai et al. [5], who proposed a new searchable encryption scheme called VSSE. Due to the fact that it is unknown that the public cloud may save computation or download bandwidth for its selfishness, the returned encrypted data may be only a fraction of all retrieval outcome. So, verifiable search-ability as well as protection of data confidentiality is a real applicable scenario during fuzzy search on cloud encrypted data. Wang et al. [19] has found the combination of fuzzy keyword search and verifiable keyword search on encrypted data and proposed a new scheme named VF-SSE, which means that data user can verify the correctness and completeness of returned files after the fuzzy search has already completed corresponding with a query containing a keyword of little typos. However, Buildindex phase in his scheme is conducted by data owner using wildcard-based scheme, which means that data owner may abandon the exact-keyword index constructed before and generate a specialized fuzzy-keyword index for fuzzy searching, thus it is inevitable for data owner to waste much more computation and storage resources. Another issue in his scheme is that ranking of keyword-retrieval has not been well tackled, and his work is mainly on the public cloud setting without applying in the hybrid cloud circumstances.

Based on thorough analysis on existing fuzzy keyword search schemes, we propose a novel scheme totally different from previous work. In this paper, we mainly concentrate on verifiable fault-tolerant keyword search on the cloud encrypted data and suggest a solution, which is called Distributed Fault-tolerant Keyword Search Supporting Verifiable Search-ability (DFKSSVS), to build secure exact-keyword index supporting verifiability in the public cloud, as well as generate fuzzy keyword trapdoors for matching in the private cloud. Due to Li's scheme weakness of insecurity, we abandon the scheme of directly using wildcard-based method to construct secure index, but we adopt traditional exact-keyword searching scheme. Our scheme will reduce index generation and storage complexity and guarantee highly efficient retrieval, and it can make fullest use of computation and storage resources in the private cloud. Our contributions of this paper can be summarized as follows:

1) We propose a novel **DFKSSVS** scheme in the hybrid cloud. We define the system and threat model, which means to be "semi-honest-but-curious" in the public cloud, and "honest-but-curious" in the private cloud. Preliminaries have been denoted to depict **DFKSSVS** scheme in detail.

2) In the public cloud, we use the exact-keyword index, which is already built for exact keyword search, or build exact-keyword index for searching for its first time. To reach verifiable search-ability, we use trie-tree based on symbol set, where a multi-way tree is constructed for storing a certain keyword trapdoor which can be recovered from the root node to the leaf node. Updating of index can be easily done on the tree structure according to trapdoor revising requests. Exact keyword search throughout the index can be well done, and encrypted data stored can be returned as well.

3) In the private cloud, we make use of the potential computation and storage resources to generate fuzzy keyword set, and trapdoors corresponding with ex-

act keywords responsible for searching on the exact-keyword index in the public cloud. At the same time, we allow statistically dynamic ranking of exact elements through feedback scheme in order to return more encrypted files related to data user's input keyword. Also, decryption of returned encrypted files is conducted and completed in the private cloud and it outputs plain-text files to data user.

The remainder of the paper is organized as follows: Section 2 introduces the system model, treat model, design goals, and preliminaries. Section 3 presents the novel scheme **DFKSSVS** in detail. Section 4 gives the security analysis of the whole scheme. Section 5 gives performance evaluation compared with [5, 11, 19] respectively. Related work for searchable encryption SE is discussed in Section 6. Finally, Section 7 concludes the paper.

## 2 Problem Statement

### 2.1 System Model

In the paper, we consider a cloud setting consisting of the entities: Data User (DU), Data Owner (DO), the Public & Private Cloud (PC), as is illustrated in Figure 1. Given a collection of $n$ files denoted as $F$ and their encrypted forms denoted as $C$, exact keyword set $W$ extracted from $F$, secure index for $C$ derived from $W$, the private cloud can generate fuzzy keyword set as well as trapdoors, which are produced with the secret key generated by authorized DUs, of similar keywords for exact matching in the public cloud, and the private cloud receives encrypted files corresponding with trapdoors, decrypts and returns the plain-text form of them to DU. Here, we denote that the private cloud can provide as much computation capability as possible just for relieving DU's burden of computing and storage. The public cloud, which is responsible for mapping trapdoors to encrypted files indexed by their IDs and linked to a series of exact keywords, supports exact matching throughout the secure index and returns encrypted files to the private cloud, and it has verifiable search-ability due to DU's verifying request to the series of searching procedures. DO has files needed to be outsourced to the cloud, and generates secret keys through $Setup(k)$ phase, which are shared with authorized DUs. DU raises a query and verifies the correctness and completeness by $proof$ sent from the public cloud. In all, our **DFKSSVS** scheme makes the fullest use of specific characters of different parties in the hybrid cloud and certainly applies the real circumstance of keyword search over a large scale of cloud encrypted data.

### 2.2 Threat Model

Firstly, we assume that authentication between DO and DU has been appropriately done. To search relevant files for a certain keyword, the trapdoor, which is of encrypted form, of the given keyword must be generated so as to match items throughout the secure index in the public cloud. And DU may want to verify the completeness of retrieval results by sending requests to the public cloud. Here, we consider the private cloud to be "honest-but-curious", which means the private cloud servers honestly obey the principles of different protocols, and have the ability to learn something additionally sensitive information, at the same time, the public cloud to be "semi-honest-but-curious", which means the public cloud servers may be selfish in order to save computation and bandwidth of its own, and have the same basic characters of private cloud servers. In addition, we take Known Cipher-text Model into consideration, which means the cloud can only have access to encrypted files, secure index and trapdoors, without leaking any information but search pattern and access pattern. The semantic meaning of the model with its proven-security has been proposed in [7].

### 2.3 Design Goals

To enable normally searching on encrypted data when typos occur, we need to do some work in the trapdoor generation procedure in order to match corresponding items throughout the index tree which is already constructed before. Furthermore, the exact-keyword index can support verifiable-searching, so we choose the basic idea of trie-tree index based on symbol tree proposed in [18], and we utilize it in the exact-keyword searching circumstance. Specifically, we have the following goals:

1) Cipher-text search supporting fault-tolerant keyword-based query: this is the basic problem to which the paper is referred, fuzzy keyword search on secure exact-keyword index constructed before is always supported as well.

2) Verifiable-searching in the cloud: this is the need of DU who wants to verify the correctness and completeness of retrieval results of a given input keyword by the $proof$ returned from the cloud.

3) Keyword privacy: in spite of leak of search pattern, the cloud should not deduce any sensitive information through secure index, encrypted trapdoor, and encrypted files, as is requested to be securely encrypted to minimize information leak risks.

4) Privacy guarantee: encrypted files should be returned to DU if and only if the correct trapdoor of a given keyword generated by authorized DU with the secret key matches the items in secure index and file IDs are obtained to link with the encrypted files.

5) Result accuracy: By feedback scheme can exact keywords in the fuzzy keyword set achieve ranking dynamically in the private cloud, which is helpful with trapdoor matching with its ranking position forward on the score list. This is similar with statistical methods, but it will not leak no more information than search pattern and access pattern by encryption.

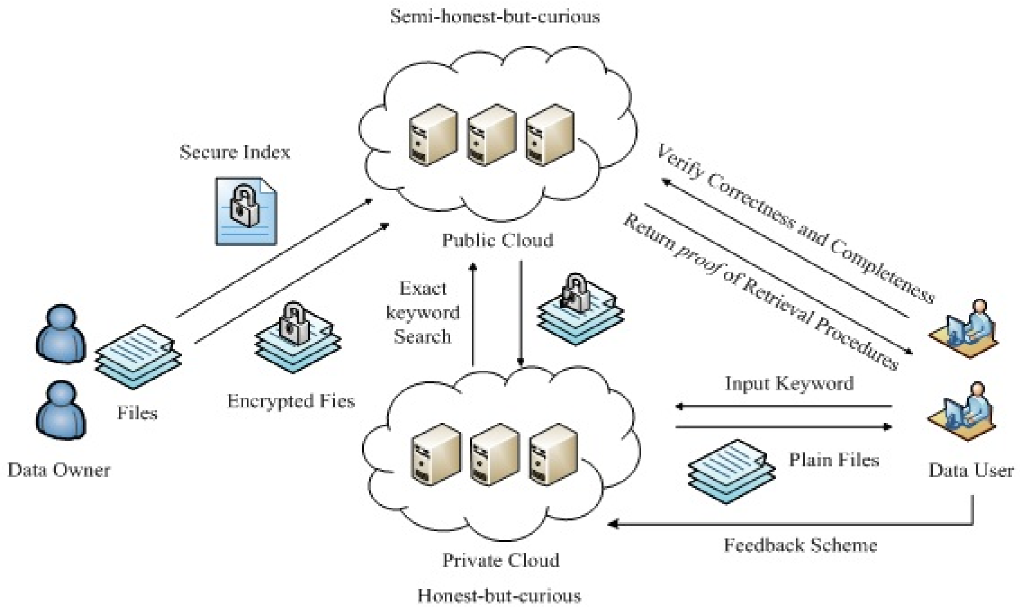


Figure 1: Architecture of system & threat model

## 2.4 Preliminaries

**Edit distance:** Given two strings $S_i$ and $S_j$, the edit distance between them is defined as $ED(S_i, S_j)$, which means the minimum steps from one string to another, including insertion, deletion, and substitution of some character in the string [13].

**Trie-tree:** A trie-tree, which is always described as prefix-tree, is a data structure with its essential order of contents in each node storing associative array where keys are always strings. Different from a binary search tree, the node of trie-tree in each position presents the key of an array, sharing the same prefix stored in its parent's node. And the root node is always associated with an empty string regarded as the starting point to conduct searching.

**Hash function:** A hash function is a function used to map any arbitrary size of data to a fixed size, with slightly different input giving rise to big difference of its output. Here in this paper, we use collision-resistant hash function (MD5, SHA-1) to generate trapdoors of exact keywords in the fuzzy keyword set, and to be the main function in the preprocessing procedure before constructing secure index outsourced to the public cloud.

**Dictionary:** it is a pre-defined keyword collection which is consisted of all indexed items (keywords) linked with certain encrypted files.

**Typical algorithms:** our scheme **DFKSSVS** is composed of six polynomial-time algorithms denoted as $KeyGen(1^k)$, $BuildIndex(sk,\ W_i)$, $ExactTrapGen(sk,\ W_i)$, $Test(I,\ Trapdoor)$, $Verify(I,\ proof)$, $Feedback(sk,\ W_s^{''})$. More details of algorithms are described below.

# 3 Distributed Fault-tolerant Keyword Search Scheme Supporting Verifiable Search-ability

In this part, we describe more details on **DFKSSVS** scheme. Based on Wang's [19] and Chai's [5] schemes proposed before, we present our novel scheme getting greater effects supported by experiential analysis and experiments on real-world data set.

## 3.1 $KeyGen(1^k)$

In the process, DO generates secret keys for index and trapdoor generation, as well as the key for keyed hash function, and file encryption. The $KeyGen$ phase is a randomized key generation algorithm, which is set up and outputs keys in the way: $hk, tk, fk \overset{R}{\longleftarrow} \{0,1\}*$, that is to say, we take $k$ as input and different secret keys are output.

## 3.2 $BuildIndex(sk, W_i)$

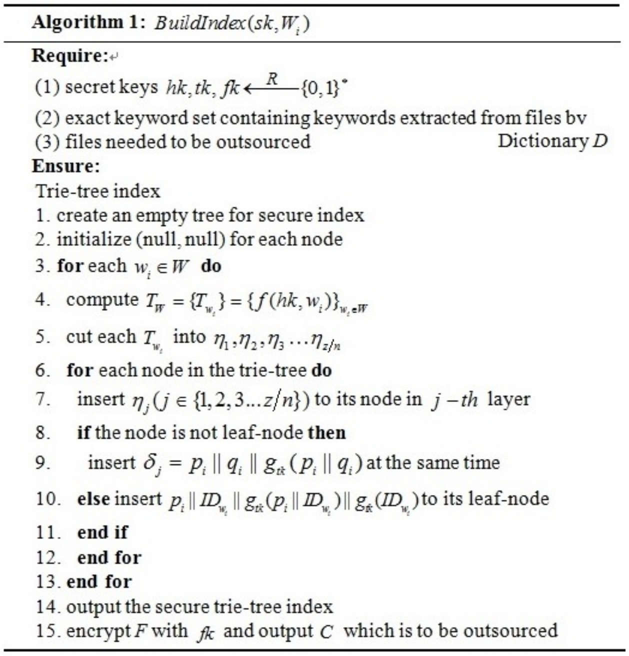
In the process, we consider to use the symbol-based trie-tree to construct secure index $I$ for the whole encrypted files of DOs. We use the scheme proposed by Chai [5] to achieve verifiable-searching on the index tree and integrate Li's [11] fuzzy keyword generation method to generate symbol-based trie-tree index based on exact keywords without abandon of exact-keyword index tree already constructed before. That is to say, we can use the new scheme to complete fuzzy searching over cloud encrypted data on the generally secure index tree without verifiability of retrieval results, or on the trie-tree index supporting verifiability of matching procedure, and we can also achieve exact keyword searching on encrypted data as well.

- Keyword extraction from files.
  In this phase, DO extracts distinct keywords from plain-text files, which are used for constructing index for each file. Different keywords have their own keyword-weight in each file and one can distinguish a specific file from others by keywords that the file possesses. Here, we denote $W$ as exact keywords extracted from the file set. We can also utilize the comprehensive dictionary $D$ to check each element's correctness and completeness of the exact-keyword set.

- Build trapdoors of exact keywords.
  To exact-keyword set, we need to generate trapdoors of elements of the keyword set by a pseudo-random one-way function, which is always used keyed hash function $f_{hk}(\cdot)$. DO computes $T_w = \{T_{w_i}\} = \{f(hk, w_i), w_i \in W\}$ for each $w_i \in W$ with the index generation key $hk$. And then, DO divides each $T_{w_i}$ into a series of $n$-length bits determined by its corresponding symbol in $\Delta$, which is denoted as $\eta_1, \eta_2, \cdots, \eta_{z/n}$, where $z$ is the output length of keyed hash function.

- Initialization of trie-tree index.
  Firstly, DO takes a quick scan of an empty trie-tree and determines that the root node is associated with an empty set as the beginning of searching throughout the whole tree. In addition, DO defines the identifier for every document file which is to be outsourced and obtains the identifier set $ID = \{ID_p, P = |F|\}$, as well as $ID_{w_i}$ representing all file IDs containing $w_i$, which is a vital path to search for the relevant encrypted files corresponding with $w_i$.

- Build symbol-based trie-tree index.
  In this phase, we mainly focus on how to insert $\eta_1, \eta_2, \cdots, \eta_{z/n}$ into the trie-tree to achieve index construction of exact keywords in the set, as well as verifiability of retrieval results. To the root node, we define an empty set to be regarded as the beginning of keyword-searching. In every child node, we insert a two-tuple unit, one is $\eta_i$ and the other is $\delta_i$, which symbolizes the route from its parent node to its own and from itself to its child node. The content of $\delta_i$ is presented as $p_i||q_1||g_{ik}(p_i||q)$, where $p_i$ is a bit-stream, which represents information of its parent node, of $2^m$ in which $\eta_i$ corresponding with the position in $\Delta$ is set to 1 while other positions are set to 0, $q_i$ represents information of its child node with the same way as mentioned above. $g_{ik}$ is a keyed hash function to encrypt node information to support verifiable-searching. For example, if the current node stores $\eta_a$, whose parent node stores $\eta_b$ with its position being the $b$-th symbol in $\Delta$, and child node stores $\eta_c$ in the same way. Then, $p_i = 0, 1, 0, 0, \cdots, 0$, $q_i = 0, 0, 1, 0, \cdots, 0$. More detailed information in all is depicted in Algorithm 1.



```
Algorithm 1: $BuildIndex(sk, W_i)$
Require:↵
(1) secret keys $hk, tk, fk \xleftarrow{R} \{0,1\}^*$
(2) exact keyword set containing keywords extracted from files bv
(3) files needed to be outsourced                    Dictionary $D$
Ensure:
Trie-tree index
create an empty tree for secure index
initialize (null, null) for each node
for each $w_i \in W$ do
  compute $T_W = \{T_{w_i}\} = \{f(hk, w_i)\}_{w_i \in W}$
  cut each $T_{w_i}$ into $\eta_1, \eta_2, \eta_3 \ldots \eta_{z/n}$
  for each node in the trie-tree do
    insert $\eta_j (j \in \{1, 2, 3 \ldots z/n\})$ to its node in $j-th$ layer
    if the node is not leaf-node then
      insert $\delta_j = p_i \| q_i \| g_{tk}(p_i \| q_i)$ at the same time
    else insert $p_i \| ID_{w_i} \| g_{tk}(p_i \| ID_{w_i}) \| g_{fk}(ID_{w_i})$ to its leaf-node
    end if
  end for
end for
output the secure trie-tree index
encrypt $F$ with $fk$ and output $C$ which is to be outsourced
```

- Files to be encrypted and outsourced.
  In this phase, files in the collection need to be encrypted by secret key $fk$ and outsourced to the public cloud. The connection between encrypted files and their IDs should be well done in the cloud so as to retrieve relevant encrypted files back to DU by searching $ID_{w_i}$ stored in the leaf node on the trie-tree index.

### 3.3 $ExactTrapGen(sk, W_i)$

In this phase, we discuss the issue of trapdoor generation corresponding with the input keyword when typos and minor mistakes occur at the beginning of the query. Considering that Li's fuzzy keyword set construction scheme has been proven weakness of its insecurity by Zheng et al. [20] in HPCC 2013 conference, which is due to mutual dependency of retrieval history, our scheme uses improved wildcard & dictionary-based construction scheme to generate trapdoors of keywords to avoid high history dependency of trapdoor relevance of distinct keywords. Because the whole search scheme is constructed in the hybrid-cloud, we assume that much work referring to some sensitive information can be done in the private cloud so as to make the fullest use of its scalable computing and storage resources even if it seems to be "honest-but-curious" for DU. We consider that plain-text keywords and secret keys are deleted after construction of exact keyword set as well as generation of exact keyword trapdoors, which means that

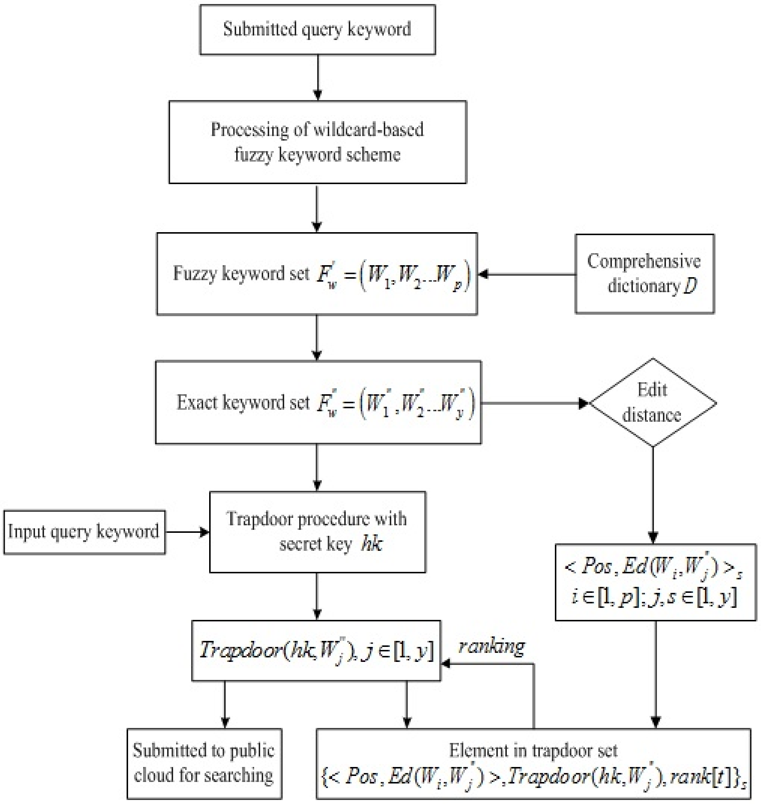

Submitted query keyword
Processing of wildcard-based fuzzy keyword scheme
Fuzzy keyword set $F'_w = (W_1, W_2 ... W_p)$
Comprehensive dictionary $D$
Exact keyword set $F''_w = (W''_1, W''_2 ... W''_y)$
Edit distance
Input query keyword
Trapdoor procedure with secret key $hk$
$< Pos, Ed(W_i, W''_j) >_s$ $i \in [1,p]; j,s \in [1,y]$
$Trapdoor(hk, W''_j), j \in [1,y]$
ranking
Submitted to public cloud for searching
Element in trapdoor set $\{< Pos, Ed(W_i, W''_j) >, Trapdoor(hk, W''_j), rank[t]\}_s$

Figure 2: Exact-keyword trapdoor construction scheme

only encrypted forms of query information can be seen in the private cloud. Figure 2 is the exact-keyword trapdoor construction scheme.

In this process, we first build the fuzzy keyword set according to the submitted query by using Li's wildcard-based fuzzy keyword construction scheme to minimize the scale of its set, then through dictionary-based method, we can determine appropriate fuzzy keywords by substituting the wildcard with fixed alphabet, so we change fuzzy search into exact search so that keyword search on encrypted data can be completed in exact-keyword search scheme. Through comparison of edit distance, edit distance information with wildcard position $Pos$ as its form of $< Pos, Ed(W_i, W'_j) >_s$ can be well calculated to be an important part of ranking tuples which will be described below. Exact keywords in the fuzzy set will be transformed into trapdoors with secret key $hk$, which are submitted to the public cloud to conduct exact matching on secure index. We use improved dictionary-based fuzzy keyword construction scheme to expand the number of keywords in the set so as to absorb much more likely exact keywords for search, which can improve the probability of relevant encrypted files needed by DU. In addition, the private cloud sends $\{Trapdoor(hk, W'_j)[\eta'_1, \eta'_2, \cdots, \eta'_{z/n}]\}_{j \in [1,y]}$ which is generated in the public cloud back to DU so as to conduct $Verify(I, proof)$ procedure.

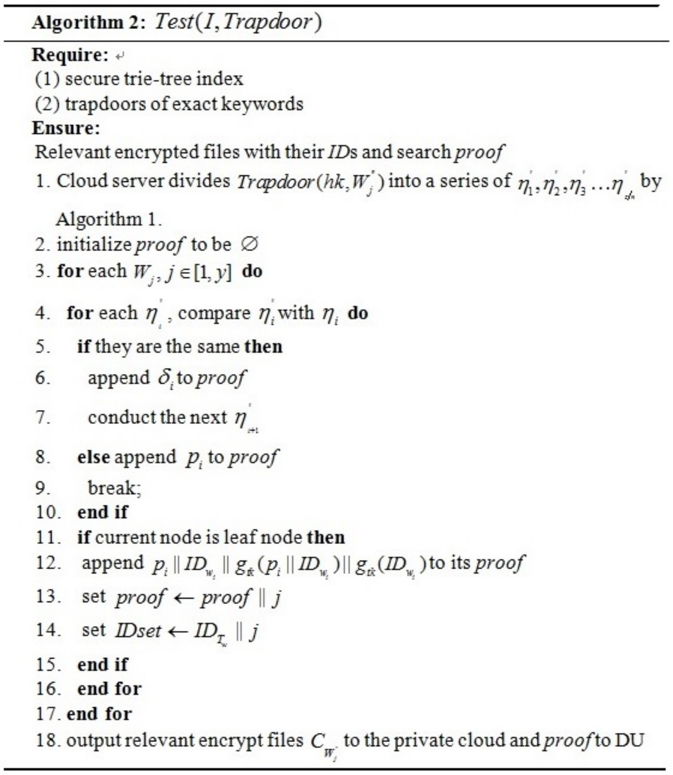
### 3.4 $Test(I, Trapdoor)$

Upon receiving search request from the private cloud, the public cloud server divides each trapdoor into a series of symbols in the same way mentioned in Algorithm 1. Then, Algorithm 2 can generate verification $proof$ containing $ID_{w_i}$ back to DU, and returns relevant encrypted files to the private cloud. According to $ID_{w_i}$, DU can check the plain-text files output by the private cloud and make requests of verification of retrieval results to the public cloud. Detailed information is shown in Algorithm 2.

### 3.5 $Verify(I, proof)$

In this part, we introduce the verification process of retrieval results in detail. We have noticed that the secret key $tk$ plays a very important role in the construction phase of $\delta_j$. Given that each node in the trie-tree has a unique route from the root node to itself, we believe that we can verify correctness and completeness of results through the $\delta_j$, which consists its unique parent node information and each child node symbol information. DO generates secret keys shared with authorized DUs, which makes attackers unable to forge a search $proof$ without the correct $tk$. And DU can verify retrieval results by re-generating $proof$ with shared secret key $tk$.

When the public cloud server completes the search pro-



**Algorithm 2:** $Test(I, Trapdoor)$

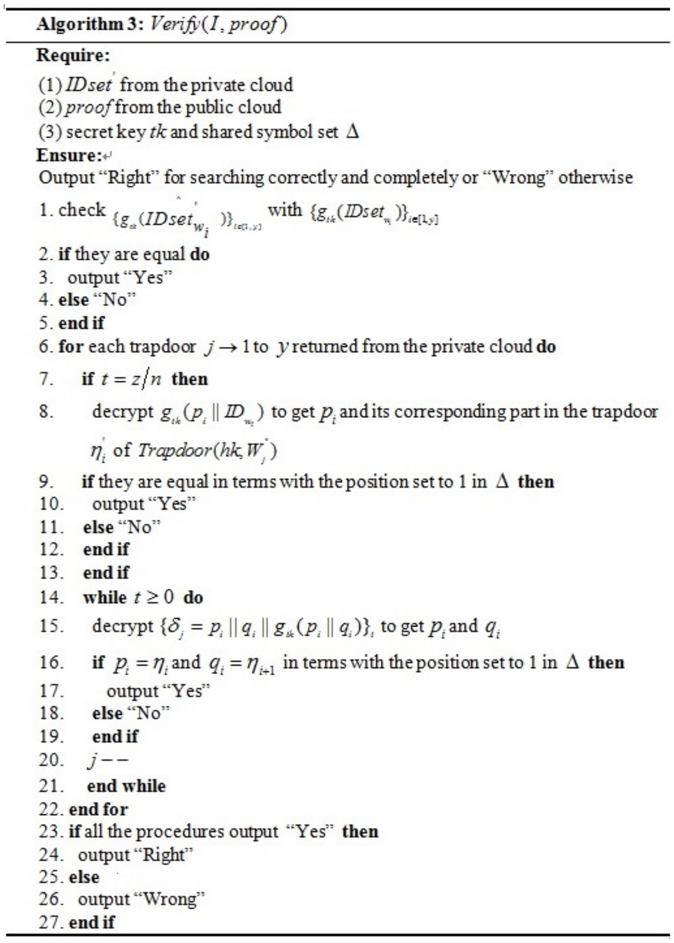
```
Require:
(1) secure trie-tree index
(2) trapdoors of exact keywords
Ensure:
Relevant encrypted files with their IDs and search proof
Cloud server divides $Trapdoor(hk, W_j^{*})$ into a series of $\eta_1^{'}, \eta_2^{'}, \eta_3^{'} \dots \eta_{d/n}^{'}$ by
    Algorithm 1.
initialize proof to be $\varnothing$
for each $W_j, j \in [1, y]$ do
  for each $\eta_i^{'}$, compare $\eta_i^{'}$ with $\eta_i$ do
    if they are the same then
      append $\delta_i$ to proof
      conduct the next $\eta_{i+1}^{'}$
    else append $p_i$ to proof
      break;
    end if
    if current node is leaf node then
     append $p_i \| ID_{w_i} \| g_{tk}(p_i \| ID_{w_i}) \| g_{tk}(ID_{w_i})$ to its proof
     set $proof \leftarrow proof \| j$
     set $IDset \leftarrow ID_{T_w} \| j$
    end if
  end for
end for
output relevant encrypt files $C_{W_j^{*}}$ to the private cloud and proof to DU
```

cess, $IDset$ corresponding with relevant encrypted files, which are the file set of returned encrypted files from the public cloud, can be obtained by DU from the private cloud to check the integrity of retrieval results, and search $proof$ is sent by the public cloud server. If the search process completes, DU can verify the results by $IDset$; otherwise, verification can be well done by $proof$ contents wherever the search process is suspended. Another point needed to be noted is that the comprehensive symbol set $\Delta$ is shared with authorized DUs. See Algorithm 3 for more detailed verifying process.

### 3.6 $Feedback(sk, W_s^{"})$

In this part, we mainly focus on feedback scheme to construct a dynamic ranking list of trapdoors without leaking of sensitive information other than search pattern and access pattern in the private cloud. Considering that the private cloud always refers to cloud service for a specific organization or government, we believe that it is less "honest-but-curious" than what we assume to be in the common sense. So we can use some plain-text information corresponding with trapdoors of exact keywords to achieve dynamic ranking of retrievals statistically from DUs. By feedback scheme can DUs receive much more relevant files containing the exact keywords derived from the input keyword with minor typos, which is benefited from the effectively statistical tendency of typos or little mistakes between DUs and the private cloud. Figure 3 is a procedure of feedback scheme in **DFKSSVS**.

We have taken it into consideration that we should not



**Algorithm 3:** $Verify(I, proof)$

```
Require:
(1) $IDset^{'}$ from the private cloud
(2) proof from the public cloud
(3) secret key tk and shared symbol set $\Delta$
Ensure:
Output "Right" for searching correctly and completely or "Wrong" otherwise
check $\{g_{tk}(\hat{IDset}^{'}_{w_i})\}_{i\in[1,y]}$ with $\{g_{tk}(IDset_{w_i})\}_{i\in[1,y]}$
if they are equal do
  output "Yes"
else "No"
end if
for each trapdoor $j \rightarrow 1$ to $y$ returned from the private cloud do
   if $t = z/n$ then
     decrypt $g_{tk}(p_i \| ID_{w_i})$ to get $p_i$ and its corresponding part in the trapdoor
     $\eta_i^{'}$ of $Trapdoor(hk, W_j^{*})$
   if they are equal in terms with the position set to 1 in $\Delta$ then
     output "Yes"
   else "No"
   end if
   end if
   while $t \geq 0$ do
     decrypt $\{\delta_j = p_i \| q_i \| g_{sk}(p_i \| q_i)\}_t$ to get $p_i$ and $q_i$
     if $p_i = \eta_i$ and $q_i = \eta_{i+1}$ in terms with the position set to 1 in $\Delta$ then
       output "Yes"
     else "No"
     end if
    $j--$
    end while
end for
if all the procedures output "Yes" then
  output "Right"
else
  output "Wrong"
end if
```

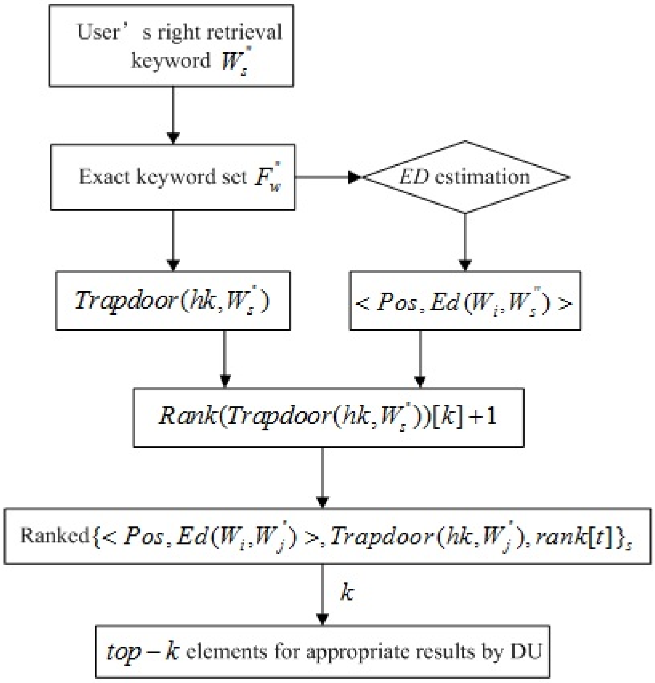

User' s right retrieval keyword $W_s^*$
Exact keyword set $F_w^*$
$ED$ estimation
$Trapdoor(hk, W_s^*)$
$< Pos, Ed(W_i, W_s^{''}) >$
$Rank(Trapdoor(hk, W_s^*))[k]+1$
Ranked$\{< Pos, Ed(W_i, W_j^*) >, Trapdoor(hk, W_j^*), rank[t]\}_s$
$k$
$top-k$ elements for appropriate results by DU

Figure 3: Procedure of feedback scheme in **DFKSSVS**

leak any sensitive information other than search pattern and access pattern, let alone the plain-text keywords existing in the private cloud. But we also emphasize the cloud being powerfully capable of computing and storage, so we choose exact keyword set with its trapdoors generation to be under control and they are to be deleted after the ranking element with its three tuples has been generated in the set. Above all, the search procedure is conducted by its encrypted form in the whole phase. Without loss of generality, we can find that the security of the novel search scheme on encrypted data has the same level as traditional SSE schemes researched before and achieves dynamically ranking of retrieval results between DUs and the cloud.

## 4 Security Analysis

**Privacy-preservation:** In this paper, we only take privacy-preserving concerns into account during the whole search procedures. For sensitive files, traditional encrypted algorithms can guarantee their security and integrity, which is out of discussion of our research. We mainly focus on confidentiality of the index and trapdoors in phases of $BuildIndex(sk, W_i)$, $ExactTrapGen(sk, W_i)$, and $Feedback(sk, W_s^{''})$. Due to security of collision-resistant keyed hash functions, generation of trapdoors can be securely conducted, which means that it is impossible for any attacker to get plain-text sensitive information without secret key $hk$ and hash algorithms. In addition, in $Feedback(sk, W_s^{''})$ phase, although we expose plain-text keywords to the private cloud so as to exploit its potentially tremendous computing capability, we can assure that risks of privacy security can be reduced to its minimum point due to timely deletion operations of exact keywords in the private cloud, and lower security threat level compared with the public cloud as well. To gain dynamical ranking of retrieval results from DUs, we make the fullest use of the private cloud to compute exact keywords corresponding with the input fuzzy keyword and trapdoors of them with a little bit sacrifices of keyword privacy in the private cloud. Here, we refer to Li's scheme security analysis to prove our scheme's search security.

**Theorem:** the novel scheme is secure regarding to its search privacy.

Similarly with Li's method, we assume that the proposed scheme cannot achieve the index & trapdoor privacy against in-distinguish-ability under chosen-plaintext-attack (IND-CPA), which means that there exists a polynomial-time algorithm $A$ who can intelligently deduce and rightly obtain plaintext sensitive information through the encrypted forms of keywords. Then, we construct another algorithm $A'$ which utilizes $A$ to decide whether $f'(\cdot) < g'(\cdot) >$ is a pseudo-random function the same as $f(sk, \cdot) < g(tk, \cdot) >$ or a real-random function. $A'$ can have access to an oracle $O_{f'(\cdot)} < O_{g'(\cdot)} >$, and takes as a real number value $x$ as input and $f'(x) < g'(x) >$ as the output. For any request of index & trapdoor generation, $A'$ can answer it with $f'(\cdot) < g'(\cdot) >$. The

Table 1: Comparison of the aforementioned schemes

| Content | Li's Scheme | Chai's Scheme | Wang's Scheme | Our scheme |
|---|---|---|---|---|
| Storage cost | $O(MN)$ | $O(N)$ | $O(MN)$ | $< O(MN)$ |
| Search cost | $O(1)$ | $O(L)$ | $O(1)$ | $O(1)$ |
| Construction cost | $O(MN)$ | $O(N)$ | $O(MN)$ | $O(1)$ |
| Verifiable-searching | NO | YES | YES | YES |
| Fuzzy-searching | YES | NO | YES | YES |
| Verification cost | - | $O(L)$ | $O(1)$ | $O(1)$ |
| Supporting ranking | - | - | - | YES |
| Search intelligence | - | - | - | YES |

adversary makes a request of two challenge keywords $w_0$ and $w_1$ after several queries to $O_{f'(\cdot)} < O_{g'(\cdot)} >$. $A'$ takes a random $b \in \{0,1\}$ and submits $w_b$ to the challenger for computing $f'(w_b) < g'(w_b) >$. Once $A'$ receives the answer $y$, it sends $y$ to the adversary to conduct a guess of answering the value of $b' \in \{0,1\}$. If $A$ can get the right value of $b'$ which is equal to $b$, $f'(w_b) < g'(w_b) >$ is not a random value, which can be directly deduced or easily guessed. By this way can $A'$ decide whether $f'(\cdot) < g'(\cdot) >$ is a real random function or not. However, due to the standard assumption of in-distinguish-ability of pseudo-random functions and real-random functions, $A$ can correctly guess the right value of $b'$ with probability of 1/2, which makes it clear that the previous assumption is wrong. So, any keyword with its encrypted form in the search process can impossibly leak any sensitive information to the attack, that is, the search procedure is secure. Another point to which we should pay attention is that our provable-security process is a little bit different from Li's and Wang's methods whose variables are fuzzy keywords generated by wildcard-based scheme expansion. And $g(tk,\cdot)$ makes it almost impossible for any adversary to fake $\delta_j = p_i||q_i||g_{tk}(p_i||q_i)$ in each node, as well as $p_i||ID_{w_i}||g_{tk}(p_i||ID_{w_i})||g_{tk}(ID_{w_i})$ in the leaf node in the whole $proof$.

**Verifiable-searching:** DU can verify correctness and completeness of retrieval results by $proof$ which the public cloud sends back. The $proof$ is composed of several parts corresponding with the content of each node in the searchable trie-tree index. We decrypt each part in the $proof$ and compare its content with trapdoor's symbol element in the fixed position of $\Delta$ in order to check consistency of the both parts. Without $tk$, it is impossible for DU to conduct the procedure of verification of results, which means verifiable-searching can be securely achieved by DU.

# 5 Experimental Performance Evaluation

In this part, we first compare our novel scheme with Li's [11], Chai's [5], and Wang's [19] schemes so as to clearly present advantages of the scheme proposed in this paper. Here, we denote N as the total number of distinct keywords and M as the maximum size of exact keyword set generated through the input fuzzy keyword. Table 1 shows the comparative contents of the four aforementioned schemes. Secondly, we give the performance evaluations and analyses under the real-world set experiment.

## 5.1 Performance Comparison

In the aspect of storage cost, our scheme is different from Li's and Wang's schemes, which need $O(MN)$ cost when fuzzy keyword set was constructed. Given that the improved dictionary-based fuzzy keyword set construction method is used for exact keyword expansion, distinct keywords in the fuzzy set can be selected and filtered by the comprehensive dictionary so that the storage cost is cut down to less than $O(MN)$. In addition, although our scheme's search cost is the same as Wang's to be $O(1)$, it is more effective for locating the symbol in the set because we adopt to use symbol's position in the set to represent its content when secure index is generated. Due to our scheme's transportability, it is always feasible for secure exact-keyword-index constructed before to conduct exact matching for relevant encrypted files, so the construction cost can be a constant number $O(1)$ compared with the three schemes which need more computation and storage cost to generate the new index. With regard to verification cost, Chai's scheme require L times decryption operations to accomplish verification procedure, where L is the length of input keyword for fuzzy searching, but our scheme inherits Wang's scheme to achieve $O(1)$ verification cost by calculating hash value to match with the $proof$ sent from the public cloud. Furthermore, our scheme achieves interaction between the DU and the private cloud so as to make the returned encrypted files in order accordingly to statistical circumstances of mi-

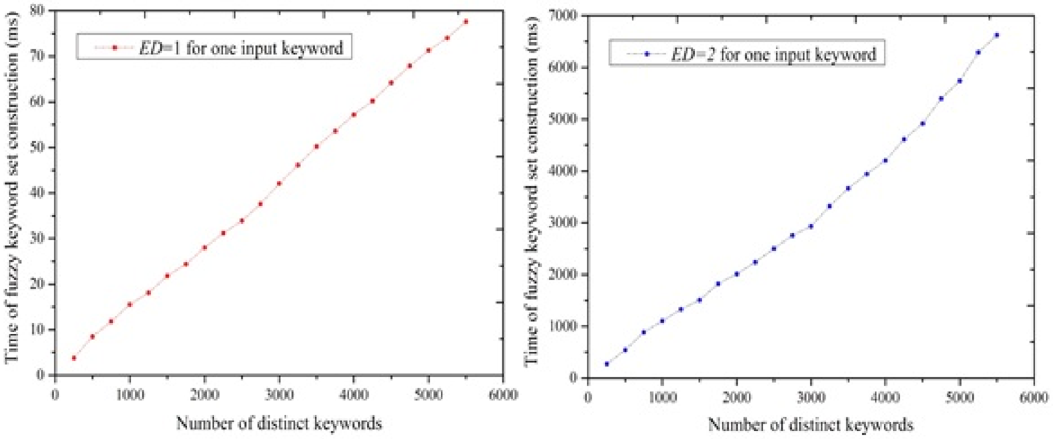


Figure 4: Time of fuzzy keyword set construction for $ED = 1$ (left) and ED=2 (right)

nor typos. This point can also achieve search intelligence through dynamic ranking of trapdoors in the private cloud in order to satisfy DU's needs of ranked retrieval results in the public cloud as well.

## 5.2 Performance Evaluation

In this section, we analyze the efficiency and accuracy of our scheme based on the experimental results in the real-world data set, that is, Request For Comments (RFC) which contains a large distinct keywords with technical files. And all the experimental results are obtained from implementation of the aforementioned schemes using JAVA on the Linux Server with Intel Core i5 Processor 4.0 GHz and 4G memory. Experimental contents include Time of Fuzzy Keyword Set Construction, Time of Index Construction, Time of Trapdoor Construction, Time of Keyword Search, and Recall & Precision of Top-k Results Corresponding with an Input Keyword, which can be also called as Performance evaluation.

Figure 4 presents time of fuzzy keyword set construction along with the number of distinct keywords for $ED = 1$ and $ED = 2$. We can easily understand that exact keywords generated by the method illustrated in Figure 2 are almost linear with the increasing number of distinct keywords for different edit distances. In addition, due to the fact that exact keywords in the fuzzy set are greatly expanded when $ED = 2$, the time of generating the fuzzy keyword set has reached 5.675s when the number of distinct keywords reaches 5000 in our experiment, which shows that edit distance is a key factor of shaping the overall efficiency of keyword search and one can no longer tolerate to waste a few seconds to generate exact keywords for fuzzy searching.

Figure 5 shows the relationship between time of index construction and number of distinct files. Because we adopt exact keywords extracted from the plain-text files to construct index, the time cost is mainly on calculating hash values of different keywords, positioning the corresponding symbols to insert inner nodes, and integrating each node's parent and child node position information to form $\delta_j$ for verification. We select [1000, 20000] files to extract and stem keywords for generating exact-keyword index, and the construction time is linear with the increase of distinct keywords corresponding with their files. Furthermore, our scheme always works well with regard to the secure exact-keyword index constructed before for searching relevant encrypted files in the public cloud.

Figure 6 gives the detailed information of trapdoor construction time along with the increase of distinct keywords. Similarly with Figure 4, the cost of trapdoor construction time is extremely large when $ED = 2$ because of the large number of exact keywords generated in the private cloud. We respectively select [1000, 10000] distinct keywords in RFC. Although edit distance is the main factor of keyword trapdoor cost, we can also find that $ED = 1$ is statistically much more common toward only one keyword. So we can take the instance of $ED = 1$ to be the main point into consideration without concern about low efficiency resulted from larger edit distances.

Figure 7 presents the time cost of keyword search. Given the real number of exact keywords indexed in the public cloud, here we choose [1000, 10000] respectively, we find that we convert fuzzy keyword searching into exact keyword matching so that the search time in terms of one input keyword has the same character as that of several times of exact matching of keywords. For all the procedures of keyword search, efficiency of our scheme can be accepted considering the existing schemes [5, 11, 16]. Moreover, because returned files include many relevant ones indexed by distinct keywords tracing back to the input keyword, the time can reach several seconds, which means that there are more selective ones for retrieving according to DU's searching interest.

Figure 8 shows the result of the proposed scheme supporting dynamic ranking, which embodies search intelligence by feedback scheme in the private cloud. Here, we use recall rate and precision rate to evaluate the whole scheme. Recall rate is denoted as $t_p/(t_p+f_n)$ while precision rate is $t_p/(t_p+f_p)$, where $t_p$ is true positive, $f_p$ is false positive, and $f_n$ is false negative. The value of $k$ in $top-k$ selection is determined by DU according to his retrieval interest. In this experiment, we first set up the times of feedback to be 10, 50, 100, and choose the value of $k$ in [10, 55] with step-length of 5 for $top-k$ selection. As illustrated in Figure 8, the recall rate is almost flat with the lowest percentage to be 96.42% no matter which $k$ is selected, and the precision rate is markedly improved from $T = 10$ to $T = 100$ with its upper bound to be 98.57% in our experiment. Although different DUs have different retrieval interests, the interaction bridge between DU and the cloud can take DU's retrieval history with statistical typos into construction of dynamic ranking list for trapdoors of exact keywords without exposing any sensitive information other than search pattern and access pattern in the private cloud, which is a sparkling point of searching intelligence in symmetric searchable encryption (SSE) field.

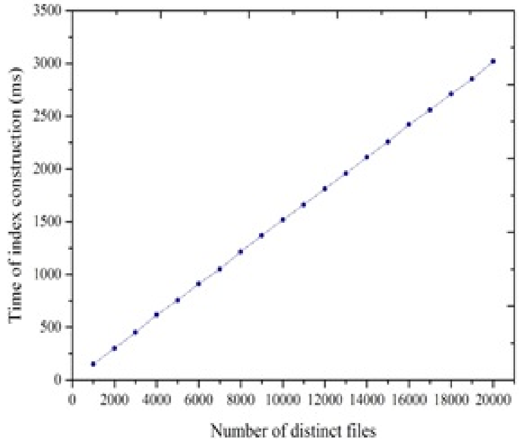


Figure 5: Time of index construction

## 6 Related Work

Song et al. [14] firstly presented the notion of searchable encryption with his specific stream cipher-text scheme. But the searching overhead is linear to the size of plaintexts which need to be encrypted. Goh [9] developed a scheme using Bloom Filter to minimize the work load under the condition of the number of all files in the collection set to establish a secure index. Boneh et al. [2] first constructed public-key based searchable encryption, whose work is so meaningful that many other scientific research teams propose different schemes achieving public-key encryption and private-key decryption. Conjunctive keyword search schemes are also recommended in [1, 3, 4, 10, 15], and specific real needs such as order-preserving symmetric encryption, single keyword or

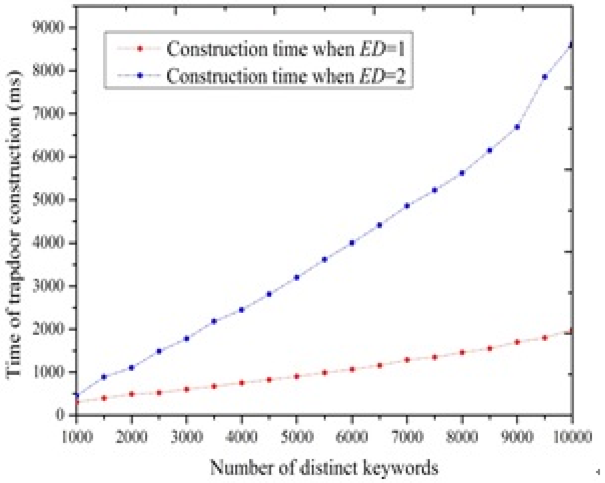


Figure 6: Time of trapdoor construction

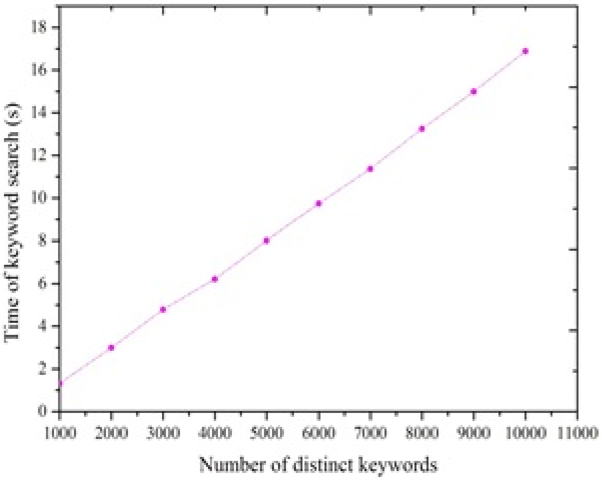


Figure 7: Time of keyword search

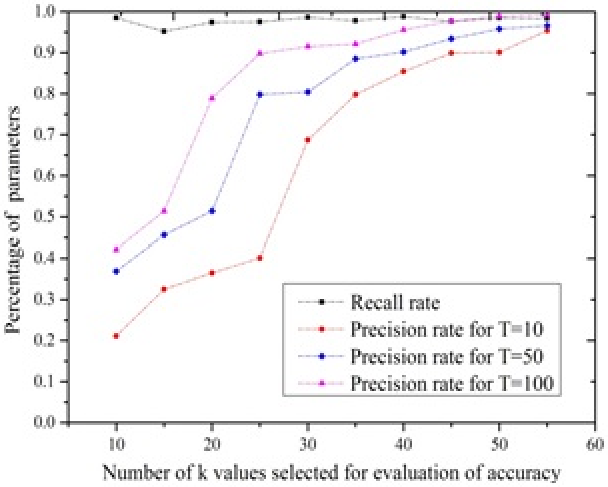


Figure 8: Performance evaluation

multi-keyword search on encrypted data and so on are likewise studied further. Application scenarios are greatly expanded due to different real work needs. Cao et al. [4], Wang et al. [17] proposed a ranked keyword search to protect privacy by using symmetric encryption schemes to achieve ranking. Different schemes are established to emphasize the keyword ranking of retrieval results in cloud encrypted data, which is the vital research direction in the searchable encryption field.

# 7 Conclusion

In this paper, we tackled the issue of fault-tolerant verifiable keyword search on cloud encrypted data. We considered the severe disadvantage of Li's scheme, which was found by Zheng presenting his conclusion in HPCC 2013 Conference, so it is important to construct a securely full-scale fuzzy keyword set so as to conduct fuzzy searching with verifiability in hybrid cloud. We proposed a novel scheme called **DFKSSVS**, which not only fully exploits infinite computing capability in the private cloud to accomplish fuzzy keyword set construction, but also supports verifiable-searching by *proof* sent from the public cloud, and dynamic ranking of retrieval results according to DU's searching history. Security analysis presents that our scheme can achieve provable-security of IND-CKA, because we have successfully converted fuzzy searching into exact keyword matching in the whole process which has been proven semantically secure before. Experimental results show accuracy, efficiency, and complexity of our new scheme, and compared with [5, 11, 19], our scheme can securely achieve our design goals—fuzzy matching, verifiable-searching, privacy-preservation, and retrieval accuracy based on dynamic ranking through feedback scheme.

# Acknowledgments

This work is supported in part by Postdoctoral Research Foundation in Aviation University of Air Force. The authors would like to thank the Northeast Area Information Data Center Workshop discussion, and show great gratitude to the anonymous reviewers proposing constructive suggestions for this paper.

# References

[1] J. Baek, R. Safavi-Naini, W. Susilo, "Public key encryption with keyword search revisited," in *Proceedings of International Conference on Computational Science and Its Applications (ICCSA'08)*, LNCS 5072, pp. 1249–1259, 2008.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search," in *Proceedings of EUROCRYPT'04*, pp. 506–522, 2004.

[3] D. Boneh, B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of the 4th Theory of Cryptography Conference*, LNCS 4392, pp. 535–554, 2007.

[4] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Privacy preserving multi-keyword ranked search over encrypted cloud data," in *Proceedings of the 31th IEEE International Conference on Computer Communications (IEEE INFOCOM'11)*, pp. 829–837, 2011.

[5] Q. Chai, G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," University of Waterloo, 2011. (http://www.cacr.math.uwaterloo.ca/techreports/2011/cacr2011-22.pdf)

[6] M. Chuah, W. Hu, "Privacy-aware bed-tree based solution for fuzzy multi-keyword search over encrypted data," in *Proceedings of the 31st International Conference on Distributed Computing Systems Workshops (ICDCSW'11)*, pp. 273–281, 2011.

[7] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient construvtions," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 79–88, 2006.

[8] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC'09)*, pp. 169–178, 2009.

[9] E. Goh, "Secure indexes," Technical Report 2003/216, Cryptology ePrint Archive, 2003. (http://eprint.iacr.org/2003/216)

[10] P. Golle, J. Staddon, B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proceedings of Applied Cryptography and Network Security (ANCS'04)*, pp. 31–45, 2004.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proceedings of the 29th IEEE International Conference on Computer Communications (IEEE INFOCOM'10)*, pp. 441–445, 2010.

[12] C. Liu, L. Zhu, L. Li, Y. Tan, "Fuzzy keyword search on encrypted cloud storage data with small index," in *Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems (IEEE CCIS'11)*, pp. 269–273, 2011.

[13] S. E. Ristad, N. Peter, "Learning string edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[14] D. Song, D. Wagner, A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy (S&P'00)*, pp. 44–55, 2000.

[15] W. Sun, B. Wang, N. Cao, M. Li, K. Ren, W. Lou, "Privacy preserving multi-keyword text search in the cloud computing supporting similarity-based ranking," in *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'13)*, pp. 71–82, 2013.

[16] B. Wang, S. Yu, W. Lou, T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proceedings of the 34th IEEE International Conference on Computer Communications (IEEE INFOCOM'14)*, pp. 1–9, 2014.

[17] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proceedings of the IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10)*, pp. 253–262, 2010.

[18] C. Wang, K. Ren, S. Yu, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *Proceedings of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM'12)*, pp. 451–459, 2012.

[19] J. Wang, X. Chen, H. Ma, Q. Tang, J. Li, H. Zhu, "A verifiable fuzzy keyword search scheme over encrypted data," *Journal of Internet Service and Information Security*, vol. 2, no. 1/2, pp. 49–58, 2012.

[20] M. Zheng, H. Zhou, "An efficient attack on a fuzzy keyword search scheme over encrypted data," in *Proceedings of 2013 IEEE International Conference on High Performance Computing and Communications (IEEE HPCC'13)*, pp. 1647–1650, 2013.

[21] W. Zhou, L. Liu, H. Jing, "K-gram based fuzzy keyword search over encrypted cloud computing," *Journal of Software Engineering and Applications*, vol. 6, no. 1, pp. 29–32, 2013.

**Jie Wang** received the MS degree in Aviation University of Air Force, China. His research interests include applied cryptography, secure cloud storage, SSE (Symmetric Searchable Encryption) as well as ASE (Asymmetric Searchable Encryption), and big-data mining. He has published more than 10 research papers in refereed domestic and international conferences and journals.

**Xiao Yu** is a post-doctor researcher at Aviation University of Air Force. He received his PhD degree in Chinese Academy in 2007. His research interests include information security and applied cryptography.

**Ming Zhao** is a professor at Aviation University of Air Force and received his doctor's degree in Jilin University in 2008. His current researches include cloud architecture, cloud storage and cloud computing security.

# Cryptanalysis of Key Exchange Method in Wireless Communication

Arindam Sarkar, Jyotsna Kumar Mandal
*(Corresponding author: Arindam Sarkar)*

Department of Computer Science & Engineering, University of Kalyani, Kalyani-741235, W.B, India
(Email: arindam.vb@gmail.com)

*(Received Oct. 5, 2012; revised and accepted Jan. 10 & July 24, 2014)*

## Abstract

In this paper, a cryptanalysis of key exchange method using multilayer perceptron (CKE) has been proposed in wireless communication of data/information. In this proposed CKE technique both sender and receiver uses an identical multilayer perceptrons for synchronization between them. After achieving the full synchronization weights vectors of both the parties' becomes identical and this identical weight vector is used as a secret session key for encryption/decryption. Different types of possible attacks during synchronization phase are introduced in this paper. Among different types of attacks some of them can be easily prevented by increasing the synaptic depth L. But few attacks are also there which has a great success rate. Parametric tests have been done and results are compared with some existing classical techniques, which show comparable results for the proposed technique.

*Keywords: Cryptanalysis, encryption, wireless communication*

## 1 Introduction

Cryptanalysis is the technique through which procedure of breaking the security can be analysed. Eavesdroppers can be reside anywhere in the network and always try to attack on the communication. In recent times wide ranges of techniques are developed to protect data and information from eavesdroppers [4, 6, 7, 8, 9, 10, 11, 14, 15]. These algorithms have their virtue and shortcomings. For Example in DES, AES algorithms [4] the cipher block length is nonflexible. In NSKTE [6], NWSKE [7], AGKNE [8], ANNRPMS [9] and ANNRBLC [10] technique uses two neural network one for sender and another for receiver having one hidden layer for producing synchronized weight vector for key generation. Now attacker can get an idea about sender and receiver's neural machine because for each session architecture of neural machine is static. In NNSKECC algorithm [11] any intermediate blocks throughout its cycle taken as the encrypted block and this number of iterations acts as secret key. Here if n number of iterations are needed for cycle formation and if intermediate block is chosen as an encrypted block after n/2th iteration then exactly same number of iterations i.e. n/2 are needed for decode the block which makes easier the attackers life. In this paper CKE technique has been proposed to analyzed variety of attacks that can be possible in key generation phase using multilayer perceptron and also provides some way out from these attacks.

The organization of this paper is as follows. Section 2 of the paper deals with structure of multilayer perceptron. Different types of attacks in CKE have been discussed in Section 3. Complexity analysis of the technique is given in Section 4. Experimental results are described in Section 5. Analysis of the results presented in Section 6. Analysis regarding various aspects of the technique has been presented in Section 7. Conclusions and future scope are drawn in Section 8 and that of references at end.

## 2 Structure of Multilayer Perceptron

In multilayer perceptron synchronization scheme secret session key is not physically get exchanged over public insecure channel. At end of neural weight synchronization strategy of both parties' generates identical weight vectors and activated hidden layer outputs for both the parties become identical. This identical output of hidden layer for both parties can be use as one time secret session key for secured data exchange. A multilayer perceptron synaptic simulated weight based undisclosed key generation is carried out between recipient and sender. Figure 1 shows multilayer perceptron based synaptic simulation system. Sender and receivers multilayer perceptron select same single hidden layer among multiple hidden layers for a particular session. For that session all other hidden layers goes in deactivated mode means hidden (processing) units of other layers do nothing with the incoming input. Either synchronized identical weight vector of sender and receivers' input layer, activated hidden layer and output layer becomes session key or session key can be form using identical output of hidden units of activated hidden layer. The key generation technique and analysis of the technique using random number of nodes (neurons) and the corresponding algorithm is discussed in Subsections 2.1 to 2.5 in details.

Sender and receiver multilayer perceptron in each session acts as a single layer network with dynamically chosen one activated hidden layer and K no. of hidden neurons, N no. of input neurons having binary input vector, , discrete weights, are generated from input to output, are lies between -L and +L, where $i = 1, \cdots, K$ denotes the $i$th hidden unit of the perceptron and $j = 1, \cdots, N$ the elements of the vector and one output neuron. Output of the hidden units is calculated by the weighted sum over the current input values. So, the state of the each hidden neurons is expressed using Equation (1).

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j}. \quad (1)$$

Output of the $i$th hidden unit is defined in Equation (2).

$$\sigma_i = sgn(h_i). \quad (2)$$

But in case of $h_i = 0$ then $\sigma_i = -1$ to produce a binary output. Hence $\sigma_i = +1$, if the weighted sum over its inputs is positive, or else it is inactive, $\sigma_i = -1$. The total output of a perceptron is the product of the hidden units expressed in Equation (3).

$$\tau = \Pi_{i=1}^{K} \sigma_i. \quad (3)$$

The learning mechanism proceeds as follows ([8, 9]):

1) If the output bits are different, $\tau A \neq \tau B$, nothing is changed.

2) If $\tau A = \tau B = \tau$, only the weights of the hidden units with $\sigma_k^{A/B} = \tau^{A/B}$ will be will be updated.

3) The weight vector of this hidden unit is adjusted using any of the following learning rules:

**Anti-Hebbian:**

$$W_k^{A/B} = W_k^{A/B} - \tau^{A/B} x_k \theta(\sigma_k \tau^{A/B})(\tau^A \tau^B). \quad (4)$$

**Hebbian:**

$$W_k^{A/B} = W_k^{A/B} + \tau^{A/B} x_k \theta(\sigma_k \tau^{A/B})(\tau^A \tau^B).$$

**Random walk:**

$$W_k^{A/B} = W_k^{A/B} + x_k \theta(\sigma_k \tau^{A/B})(\tau^A \tau^B).$$

During Step (2), if there is at least one common hidden unit with $\sigma k = \tau$ in the two networks, then there are 3 possibilities that characterize the behavior of the hidden nodes:

1) An attractive move: if hidden units at similar $k$ positions have equal output bits, $\sigma_k^A = \sigma_k^B = \tau^{A/B}$.

2) A repulsive move: if hidden units at similar $k$ positions have unequal output bits, $\sigma_k^A \neq \sigma_k^B$.

3) No move: when $\sigma_k^A = \sigma_k^B \neq \tau^{A/B}$, the distance between hidden units can be defined by their mutual overlap,

$$\rho_k = \frac{w_k^A w_k^B}{\sqrt{w_k^A w_k^A}\sqrt{w_k^B w_k^B}}$$

where $0 < \rho k < 1$, with $\rho k = 0$ at the start of learning and $\rho k = 1$ when synchronization occurs with the two hidden units having a common weight vector.

## 2.1 Multilayer Perceptron Simulation Algorithm

**Input:** Random weights, input vectors for both multilayer perceptrons.

**Output:** Secret key through synchronization of input and output neurons as vectors.

**Method:**

**Step 1.** Initialization of random weight values of synaptic links between input layer and randomly selected activated hidden layer.

$$w_{i,j} \in \{-L, -L+1, \cdots, +L\}. \quad (5)$$

**Step 2.** Repeat Steps 3 to 6 until the full synchronization is achieved, using Hebbian-learning rules.

$$w_{i,j}^{+} = g(w_{i,j} + x_{i,j} \tau \theta(\sigma_i \tau) \theta(\tau^A \tau^B)).$$

**Step 3.** Generate random input vector $X$. Inputs are generated by a third party or one of the communicating parties.

**Step 4.** Compute the values of the activated hidden neurons of activated hidden layer using Equation (6).

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j}. \quad (6)$$

**Step 5.** Compute the value of the output neuron using

$$\tau = \Pi_{i=1}^{K} \sigma_i.$$

Compare the output values of both multilayer perceptron by exchanging the system outputs.

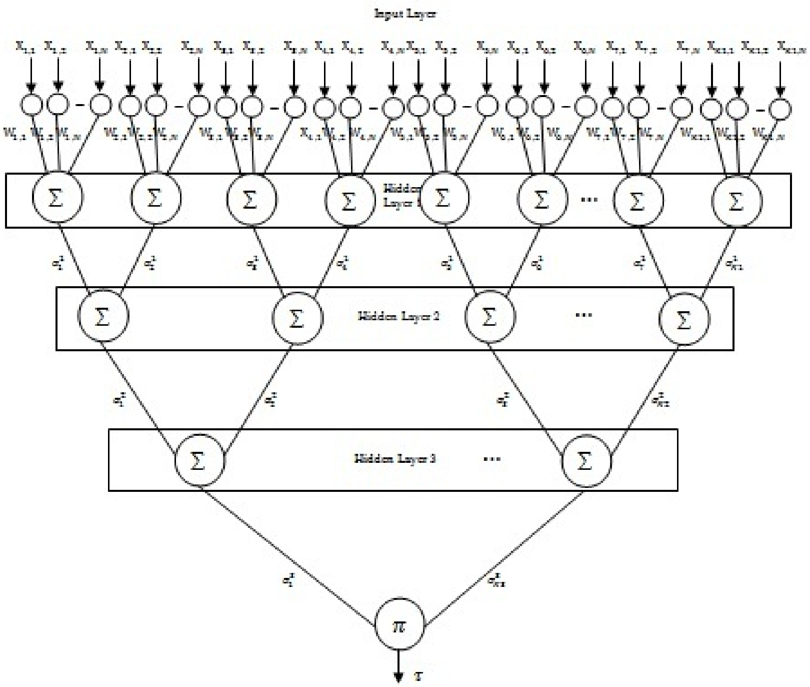


Figure 1: A multilayer perceptron with 3 hidden layers

*if Output (A) $\neq$ Output (B), Go to Step 3*

*else if Output (A) = Output (B) then one of the suitable learning rule is applied*

*only the hidden units are trained which have an output bit identical to the common output.*

Update the weights only if the final output values of the perceptron are equivalent. When synchronization is finally achieved, the synaptic weights are identical for both the system.

## 2.2 Multilayer Perceptron Learning Rule

At the beginning of the synchronization process multilayer perceptron of $A$ and $B$ start with uncorrelated weight vectors. For each time step $K$, public input vectors are generated randomly and the corresponding output bits $A/B$ are calculated. Afterwards $A$ and $B$ communicate their output bits to each other. If they disagree, $A \neq B$, the weights are not changed. Otherwise learning rules suitable for synchronization is applied. In the case of the Hebbian learning rule [12] both neural networks learn from each other.

$$w_{i,j}^{+} = g(w_{i,j} + x_{i,j}\tau\theta(\sigma_i\tau)\theta(\tau^A\tau^B)).$$

The learning rules used for synchronizing multilayer perceptron share a common structure. That is why they can be described by a single Equation (4).

$$w_{i,j}^{+} = g(w_{i,j} + f(\sigma_i, \tau^A, \tau^B)x_{i,j})$$

with a function $f(\sigma_i, \tau^A, \tau^B)$, which can take the values -1, 0, or +1. In the case of bidirectional interaction it is given by

$$\begin{aligned} & f(\sigma_i, \tau^A, \tau^B) \\ = \ & \theta(\sigma\tau^A)\theta(\tau^A\tau^B)\begin{cases} \sigma, & \text{Hebbian learning} \\ -\sigma & \text{anti-Hebbian learning} \\ 1 & \text{Random walk learning} \end{cases} \end{aligned}$$

The common part $\theta(\sigma\tau^A)\theta(\tau^A\tau^B)$ of $f(\sigma_i, \tau^A, \tau^B)$ controls, when the weight vector of a hidden unit is adjusted. Because it is responsible for the occurrence of attractive and repulsive steps [8].

The equation consists of two parts:

1) $\theta(\sigma\tau^A)\theta(\tau^A\tau^B)$: This part is common between the three learning rules and it is responsible for the attractive and repulsive effect and controls when the weight vectors of a hidden unit is updated. Therefore, all three learning rules have similar effect on the overlap.

2) $(\sigma, -\sigma, 1)$: This part differs among the three learning rules and it is responsible for the direction of the weights movement in the space. Therefore, it changes the distribution of the weights in the case of Hebbian and anti-Hebbian learning. For the Hebbian rule, A's ad B's multilayer perceptron learn their own output and the weights are pushed towards the boundaries at $-L$ and $+L$. In contrast, by using the anti- Hebbian rule, sender's and receiver's multilayer perceptron learn the opposite of their own outputs.

Consequently, the weights are pulled from the boundaries $\pm L$. The random walk rule is the only rule that does not affect the weight distribution so they stay uniformly distributed. In fact, at large values of $N$, both Hebbian and anti-Hebbian rules do not affect the weight distribution. Therefore, the proposed algorithm is restricted to use either random walk learning rule or Hebian or anti-Hebbian learning rules only at large values of $N$. The random walk learning rule is chosen since it does not affect the weights distribution regardless of the value of $N$.

## 2.3 Weight Distribution of Multilayer Perceptron

In case of the Hebbian rule Equation (5), A's and B's multilayer perceptron learn their own output. Therefore the direction in which the weight $w_{i,j}$ moves is determined by the product $\sigma_i x_{i,j}$. As the output $\sigma_i$ is a function of all input values, $x_{i,j}$ and $\sigma_i$ are correlated random variables. Thus the probabilities to observe $\sigma_i x_{i,j} = +1$ or $\sigma_i x_{i,j} = -1$ are not equal, but depend on the value of the corresponding weight $w_{i,j}$ [2, 3, 5, 13].

$$P(\sigma_i x_{i,j} = 1) = \frac{1}{2}[1 + erf(\frac{e_{i,j}}{\sqrt{NQ_i - w_{i,j}^2}})].$$

According to this equation, $\sigma_i x_{i,j} = sgn(w_{i,j})$ occurs more often than the opposite, $\sigma_i x_{i,j} = -sgn(w_{i,j})$. Consequently, the Hebbian learning rule pushes the weights towards the boundaries at $-L$ and $+L$. In order to quantify this effect the stationary probability distribution of the weights for $t \to \infty$ is calculated for the transition probabilities. This leads to [13].

$$P(w_{i,j} = w) = P_0 \prod_{m=1}^{|w|} \frac{1 + erf[\frac{m-1}{\sqrt{NQ_i - (m-1)^2}}]}{1 - erf[\frac{m}{\sqrt{NQ_i - m^2}}]}.$$

Here the normalization constant $P_0$ is given in Equation (7), the constant should be expressed as

$$P_0 = [\sum_{w=-L}^{L} \prod_{m=1}^{|w|} \frac{1 + erf[\frac{m-1}{\sqrt{NQ_i - (m-1)^2}}]}{1 - erf[\frac{m}{\sqrt{NQ_i - m^2}}]}]^{-1}. \quad (7)$$

In the limit $N \to \infty$ the argument of the error functions vanishes, so that the weights stay uniformly distributed. In this case the initial length of the weight vectors is not changed by the process of synchronization.

$$\sqrt{Q_i(t = 0)} = \sqrt{\frac{L(L+1)}{3}}.$$

But, for finite $N$, the probability distribution itself depends on the order parameter $Q_i$. Therefore its expectation value is given by the solution of the following equation:

$$Q_i = \sum_{w=-L}^{L} w^2 P(w_{i,j} = w).$$

## 2.4 Order Parameters

In order to describe the correlations between two multilayer perceptron caused by the synchronization process, one can look at the probability distribution of the weight values in each hidden unit. It is given by $(2L + 1)$ variables.

$$P_{a,b}^i = P(w_{i,j}^A = a \wedge w_{i,j}^B = b)$$

which are defined as the probability to find a weight with $w_{i,j}^A = a$ in A's multilayer perceptron and $w_{i,j}^B = b$ in B's multilayer perceptron. In both cases, simulation and iterative calculation, the standard order parameters, which are also used for the analysis of online learning, can be calculated as functions of $P_{a,b}^i$ [1].

$$\begin{aligned} Q_i^A &= \frac{1}{N} w_i^A w_i^A = \sum_{a=-L}^{L} \sum_{b=-L}^{L} a^2 P_{a,b}^i \\ Q_i^B &= \frac{1}{N} w_i^B w_i^B = \sum_{a=-L}^{L} \sum_{b=-L}^{L} b^2 P_{a,b}^i \\ R_i^{AB} &= \frac{1}{N} w_i^A w_i^B = \sum_{a=-L}^{L} \sum_{b=-L}^{L} ab P_{a,b}^i \end{aligned}$$

Then the level of synchronization is given by the normalized overlap between two corresponding hidden units

$$\rho_i^{AB} = \frac{w_i^A w_i^B}{\sqrt{w_i^A w_i^A}\sqrt{w_i^B w_i^B}} = \frac{R_i^{AB}}{\sqrt{Q_i^A Q_i^B}}.$$

## 2.5 Hidden Layer as a Secret Session Key

At end of full weight synchronization process, weight vectors between input layer and activated hidden layer of both multilayer perceptron systems become identical. Activated hidden layer's output of source multilayer perceptron is used to construct the secret session key. This session key is not get transmitted over public channel because receiver multilayer perceptron has same identical activated hidden layer's output. Compute the values of the each hidden unit by

$$\begin{aligned} \sigma_i &= sgn(\sum_{j=1}^{N} w_{i,j} x_{i,j}) \\ sgn(x) &= \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \end{aligned} \quad (8)$$

For example consider 8 hidden units of activated hidden layer having absolute value $(1, 0, 0, 1, 0, 1, 0, 1)$ becomes an 8 bit block. This 10010101 become a secret session key for a particular session and cascaded XORed with recursive replacement encrypted text. Now final session key based encrypted text is transmitted to the receiver end. Receiver has the identical session key i.e. the output of the hidden units of activated hidden layer of receiver. This session key used to get the recursive replacement encrypted text from the final cipher text. In the next session both the machines started tuning again to produce another session key. Identical weight vector derived from synaptic link between input and activated hidden layer of both multilayer perceptron can also becomes secret session key for a particular session after full weight synchronization is achieved.

# 3 Different Types of Attacks on Multilayer Perceptron

The security of multilayer perceptron based key generation protocol is based on a contest between attractive and repulsive forces. Two multilayer perceptrons interacting with each other synchronize much faster than an attacker network only trained with their inputs and outputs. The dissimilarity between the two parties and the attacker is that the two parties synchronize in a polynomial time of synaptic depth $L$, while the complexity of the attacker scales exponentially. However, the process is stochastic and depends on the random attractive and repulsive forces. As a result, there is a small probability that an attacker succeeds to synchronize with one of the parties. The difficulty an attacker faces with the organization of multilayer perceptron is the lack of information about the internal representation of A's or B's machine. Most of attacks depend on estimating the state of the hidden units. Following are the different possible attacks on multilayer perceptron during key generation phase.

## 3.1 Type-1 Attack

In this type1attack replicate a huge population of multilayer perceptrons with the identical arrangement as the two parties, and teach them with the same inputs. At each stage about half the replicated networks produces an output of $+1$, and half produces an output of $-1$. Successful multilayer perceptrons whose outputs imitate those of the two parties raise and multiply, while unsuccessful multilayer perceptrons gets ruled out. Attack starts with one network with haphazardly chosen weights. At each step a population of networks grow according to 3 potential scenarios:

- $A$ and $B$ have dissimilar outputs $\tau A \neq \tau B$, and therefore do not change their weights. Then all the attacker's networks stay unaffected as well.

- $A$ and $B$ have the equivalent outputs $\tau A = \tau B$, and the sum of attacking networks is lesser than some predefined limit. In this case there are 4 possible combinations of the hidden outputs agreeing with the final output. So, the attacker replaces each network $N$ from the population by 4 variants of itself, $\{N1, ..., N4\}$ which are the results of updating $N$ with the standard learning rule but pretending that the hidden outputs were equal to each one of these combinations.

- $A$ and $B$ have the identical outputs $\tau A = \tau B$ but the total number of simulated networks is larger than predefined value. In this case the attacker computes the outputs of all the networks, deletes the unsuccessful networks whose output is different from $\tau A$, and updates the weights in the successful networks by using the standard learning rule with the actual hidden outputs of the perceptrons.

## 3.2 Type-2 Attack

In Type-2 attack the attacker imitates one of the parties, but if attacker output disagrees with the imitated party's output $\tau c \neq \tau A$, attacker certainly knows that either one or all three of his hidden units are mistaken. In order to get $\tau c = \tau A$ attacker negates the sign of one of attacker's hidden units. As $\sigma = sgn(h)$ the unit most likely to be wrong is the one with the minimal $|h|$, therefore that is the unit which is negate. This policy results a immense enhancement in the attacker's achievement. It can be seen that the success rate is quite high for all $L$ values presented, but it drops exponentially as $L$ increases. On the other hand parties' synchronization time increases like $L^2$, and therefore it can be conclude that in the boundary of large $L$ values the proposed technique is secure against Type-2 attack. Each input can be viewed as $K$ random hyperplanes $(X_1, \cdots, X_K)$ corresponding to $K$ hidden units. Each $X_i$ is a hyperplane $f_i(z_1, \cdots, z_n) = \sum_{j=i}^{N} x_{ij} z_j = 0$ in the $N$-dimensional discrete space $U = \{-L, \cdots, L\}^N$. The weights of a network could be also viewed as $K$ points $W_1, \cdots, W_K$ in $U$, $W_i = \{w_{i1}, \cdots, w_{ik}\}$, while the $i$-th hidden output is just the side of the half-space (with respect to $X_i$) which contains $W_i$. Consider an attacking network $E$ that is close enough to the unknown network $A$ but has a different output for a given input. In fact they have either 1 or 3 different hidden outputs. The second case is less likely to occur so we assume that only one hidden output of the network $E$ is different from the corresponding hidden output of $A$. Consequently, only one pair $(W_i^A, W_I^E)$ is separated by the known input hyperplane $X_i$. Of course, we are interested in detecting its index $i$. If the points $W_i^E$ and $W_i^A$ are separated by $X_i$ then the distance between them is greater than the distance from $W_i^E$ to the hyperplane $X_i$. $W_i^E$ and $W_i^A$ are close to each other, so the distance from $W_i^E$ to $X_i$ has to be small. On the other hand, if $W_i^E$ and $W_i^A$ are in the same half-space with respect to $X_i$ then they are more likely to be far away from the random input $X_i$ (even though we know that they are close to each other). We thus guess that the index of the incorrect hidden output is the $i$ for which $W_i^E$ is closest to the corresponding hyperplane $X_i$, where we compute the distance by $\rho(W_i^E, X_i) = |f_i(W_i^E)|$. Formally, the attacker constructs a single neural network $E$ with the same structure as $A$ and $B$, and randomly initializes its weights. At each step attacker's trains $E$ with the same input as the two parties, and updates its weights with the following rules:

- If $A$ and $B$ have different outputs $\tau A \neq \tau B$, then the attacker doesn't update $E$.
- If $A$ and $B$ have the same outputs $\tau A = \tau B$ and $\tau E = \tau A$, then the attacker updates $E$ by the usual learning rule.
- If $A$ and $B$ have the same outputs $\tau A = \tau B$ and $\tau E \neq \tau A$, then the attacker finds $i_0 \in \{1, \cdots, K\}$ that minimizes $|\sum_{j=0}^{N} w_{ij}^{E} x_{ij}|$. The attacker negates $\tau_{i_0}^{E}$ and updates $E$ assuming the new hidden bits and output $\tau A$.

### 3.3 Type-3 Attack

In this Type-3 attack a huge collection of $M$ attackers work together. The Type-2 attacker's likelihood to supposition correctly A's interior representation is some function $Pcorrect(\beta)$ of its overlap $\beta$ with $A$, starting from $Pcorrect(\beta = 0) = 0.25$. Assume there are group of $M$ independent Type-2 attackers, each having overlap $\beta$ with $A$. They will split into 4 groups, one for each possible internal representation. Since $Pcorrect > 0.25$ for all $\beta > 0$, the number of attackers having the correct internal representation, $M$. $Pcorrect$ will be bigger than the number of attackers in the other 3 groups, for all $\beta > 0$. Therefore, the internal representation resulted from the majority discussion of $M$ independent Type-2 attackers would always be the correct one! From this argument we conclude that the attack should use $M \gg 2k-1$ attackers, which would simultaneously develop an overlap with the parties, trying to remain as independent as possible.

### 3.4 Type-4 Attack

The Type-4 attack procedure is to start from independent Type-2 attackers and let them act disjointedly for some preliminary number of time steps. Then, the majority procedure is applied: we count how many attackers have each of the 4 possible internal representations, and assign the majority's internal representation to all the $M$ attackers. To prevent the similarity between the attackers from developing too quickly, this majority procedure is applied only on even time steps. However, the attackers make many coherent moves, and unavoidable overlap is developed between them as well. Therefore we do not have a group of independent attackers, but of attackers with an overlap between them. This overlap diminishes the efficiency of the attack, and it is not always successful as a majority attack of $M$ independent attackers would be.

### 3.5 Type-5 Attack

It is much easier to predict the position of a point in a bounded multidimensional box after several moves in its random walk than to guess its original position. A simple way to do it is to consider each coordinate separately, and to associate with each possible value $i$ in the interval $\{-L, \cdots, L\}$ of the probability $p_t(i) = Pr[X_t = i]$. Initially $\forall i, p_0(i) = \frac{1}{2L+1}$ and after each move $p_{t+1}(i) = \sum_j p_t(j)$, where $j$ are such that if $x_t = j$ then $x_{t+1} = i$. Applying this technique to the original scheme we face the problem that the moves are not known to the attacker does not know which perceptrons are updated in each round. Fortunately, if we know the distribution of the probabilities $P_{k,n,i} = Pr[w_{k,n} = i]$ then using dynamic programming we can calculate the distribution of $\vec{w_k}\vec{x_k}$ for a given vector $\vec{x_k}$ and thus the probabilities $u_k(s) = Pr[\tau_k = s]$. Using these probabilities we can calculate the conditional probabilities

$$\begin{aligned} U_k &= Pr[\tau_k = 1|\tau], \\ &= \frac{\sum_{(\alpha_1,\cdots,\alpha_k):\prod_i \alpha_i = \tau, \alpha_k = 1} \prod_i \mu_i(\alpha_i)}{\sum_{(\alpha_1,\cdots,\alpha_k):\prod_i \alpha_i = \tau} \prod_i \mu_i(\alpha_i)} \end{aligned}$$

because $\tau$ is publicly known. We can now update the distribution of the weights: $P_{k,n,i}^{t+1} = \sum_j P_{k,n,j}^{t} Pr[w_{k,n}^{t} = j \Rightarrow w_{k,n}^{t+1} = i]$ is calculated using $U_k$. Experiments show that in most cases, when $A$ and $B$ converge to a common $\hat{w_{k,n}}$ the probabilities $Pr[w_{k,n} = \hat{w_{k,n}} \approx 1$ and thus the adversary can easily find $\hat{w_{k,n}}$ when $A$ and $B$ decide to stop the protocol.

### 3.6 Type-6 Attack

To provide a brute force attack, an attacker has to test all possible keys (all possible values of weights). By $K$ hidden neurons, $K \times N$ input neurons and boundary of weights $L$, this gives $(2L+1)KN$ possibilities. For example, the configuration $K = 3$, $L = 3$ and $N = 100$ gives us $3 \times 10253$ key possibilities, making the attack impossible with today's computer power.

### 3.7 Type-7 Attack

Here the attacker E's neural network has the same structure of A's and B's. All what $E$ has to do is to start with random initial weights and to train with the same inputs transmitted between $A$ and $B$ over the public channel. Then, the attacker $E$ learns the mutual output bit $\tau^{A/B}$ between them and applies the same learning rule by replacing $\tau^E$ with $\tau^{A/B}$, i.e.

$$W_k^E = W_k^E - \tau^{A/B} x_k \theta(\sigma_k^E \tau^{A/B})(\tau^A \tau^B).$$

One of the basic attacks can be provided by an attacker, who owns the same tree parity machine as the parties $A$ and $B$. He wants to synchronize his tree parity machine with these two parties. In each step there are three situations possible:

- Output (A) $\neq$ Output (B): None of the parties updates its weights.

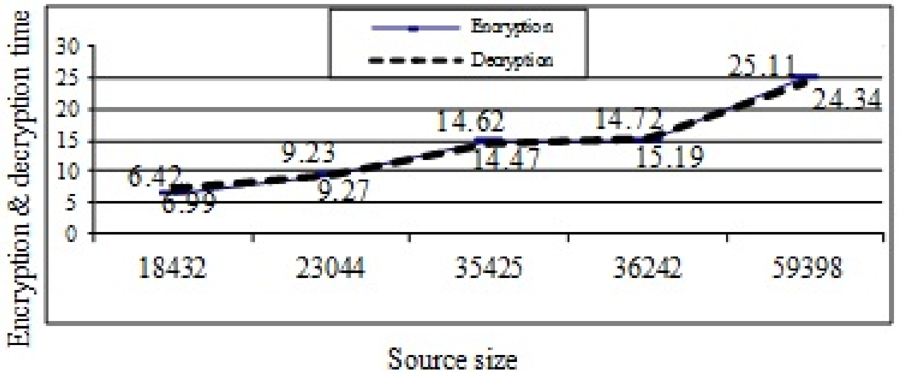


Figure 2: Encryption decryption time against stream size

- Output (A) = Output (B) = Output (E): All the three parties update weights in their tree parity machines.
- Output (A) = Output (B) ≠ Output (E): Parties $A$ and $B$ update their tree parity machines, but the attacker cannot do that. Because of this situation his learning is slower than the synchronization of parties $A$ and $B$.

It has been proven, that the synchronization of two parties is faster than learning of an attacker. It can be improved by increasing of the synaptic depth L of the neural network. That gives this protocol enough security and an attacker can find out the key only with small probability. Changing this parameter increases the cost of a successful attack exponentially, while the effort for the users grows polynomially. Therefore, breaking the security of neural key exchange belongs to the complexity class NP.

## 4 Complexity Analysis

The complexity of the Synchronization technique will be $O(L)$, which can be computed using following three steps.

**Step 1.** To generate a MLP guided key of length $N$ needs $O(N)$ Computational steps. The average synchronization time is almost independent of the size $N$ of the networks, at least up to $N = 1000$. Asymptotically one expects an increase like $O(logN)$.

**Step 2.** Complexity of the encryption technique is $O(L)$.

**Step 2.1.** Recursive replacement of bits using prime nonprime recognition encryption process takes $O(L)$.

**Step 2.2.** MLP based encryption technique takes $O(L)$ amount of time.

**Step 3.** Complexity of the decryption technique is $O(L)$.

**Step 3.1.** In MLP based decryption technique, complexity to convert final cipher text into recursive replacement cipher text $T$ takes $O(L)$.

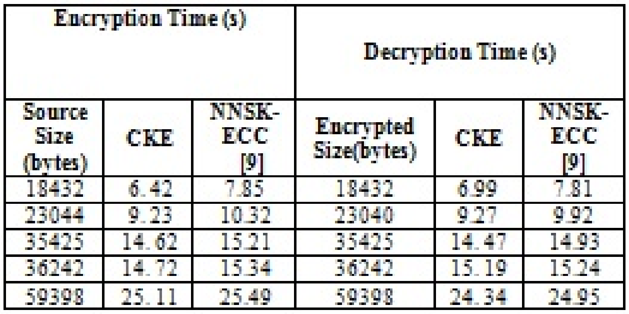
**Step 3.2.** Transformation of recursive replacement cipher text $T$ into the corresponding stream of bits $S = s_0s_1s_2s_3s_4 \cdots s_{L-1}$, which is the source block takes $O(L)$ as this step also takes constant amount of time for merging $s_0s_1s_2s_3s_4 \cdots s_{L-1}$.

## 5 Experiment Results

In this section the results of implementation of the proposed CKE encryption/decryption technique has been presented in terms of encryption decryption time, Chi-Square test, source file size vs. encryption time along with source file size vs. encrypted file size.

The results are also compared with existing RSA [4] technique, existing ANNRBLC [10] and NNSKECC [11].

Table 1: Encryption/decryption time vs. file size



| Encryption Time (s) | | | Decryption Time (s) | | |
|---|---|---|---|---|---|
| Source Size (bytes) | CKE | NNSK-ECC [9] | Encrypted Size(bytes) | CKE | NNSK-ECC [9] |
| 18432 | 6.42 | 7.85 | 18432 | 6.99 | 7.81 |
| 23044 | 9.23 | 10.32 | 23040 | 9.27 | 9.92 |
| 35425 | 14.62 | 15.21 | 35425 | 14.47 | 14.93 |
| 36242 | 14.72 | 15.34 | 36242 | 15.19 | 15.24 |
| 59398 | 25.11 | 25.49 | 59398 | 24.34 | 24.95 |

Table 1 shows encryption and decryption time with respect to the source and encrypted size respectively. It is also observed the alternation of the size on encryption.

In Figure 2 stream size is represented along X axis and encryption/decryption time is represented along Y-axis. This graph is not linear, because of different time requirement for finding appropriate CKE key. It is observed that the decryption time is almost linear, because there is no CKE key generation process during decryption.

Table 2 shows Chi-Square value for different source stream size after applying different encryption algorithms.

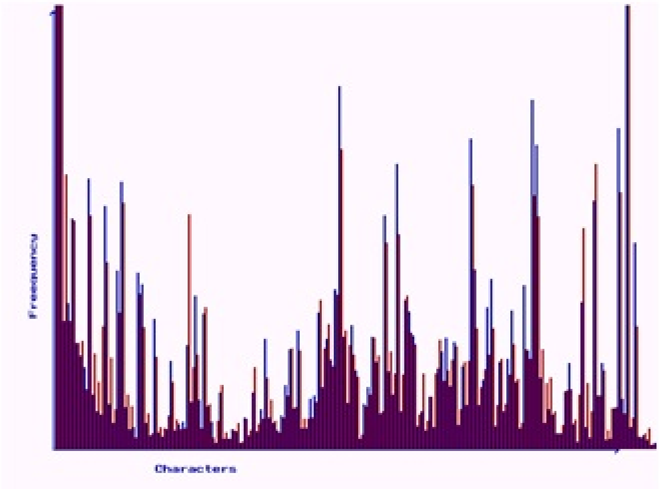
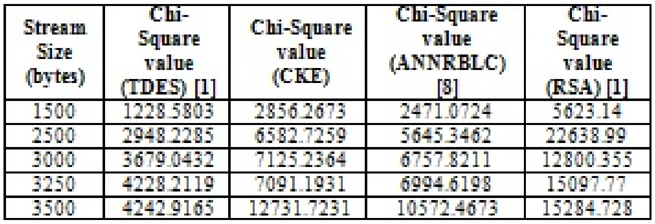


Figure 3: Chi-Square value against stream size

It is seen that the Chi-Square value of CKE is better compared to the algorithm ANNRBLC [10] and comparable to the Chi-Square value of the RSA algorithm. Figure 3 shows graphical representation of Table 2.

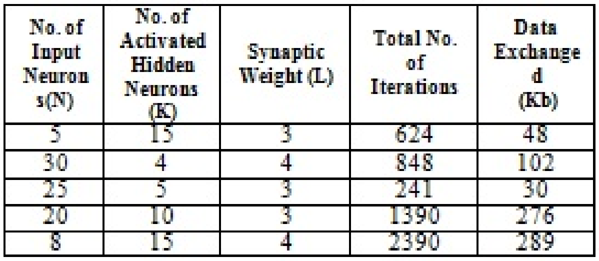
Table 2: Source size vs. Chi-Square value



| Stream Size (bytes) | Chi-Square value (TDES) [1] | Chi-Square value (CKE) | Chi-Square value (ANNRBLC) [8] | Chi-Square value (RSA) [1] |
|---|---|---|---|---|
| 1500 | 1228.5803 | 2856.2673 | 2471.0724 | 5623.14 |
| 2500 | 2948.2285 | 6582.7259 | 5645.3462 | 22638.99 |
| 3000 | 3679.0432 | 7125.2364 | 6757.8211 | 12800.355 |
| 3250 | 4228.2119 | 7091.1931 | 6994.6198 | 15097.77 |
| 3500 | 4242.9165 | 12731.7231 | 10572.4673 | 15284.728 |

Table 3 shows total number of iteration needed and number of data being transferred for CKE key generation process with different numbers of input(N) and activated hidden(H) neurons and varying synaptic depth(L). Figure 4 shows the snapshot of CKE key simulation process.

Table 3: Data exchanged and number of iterations for different parameters value

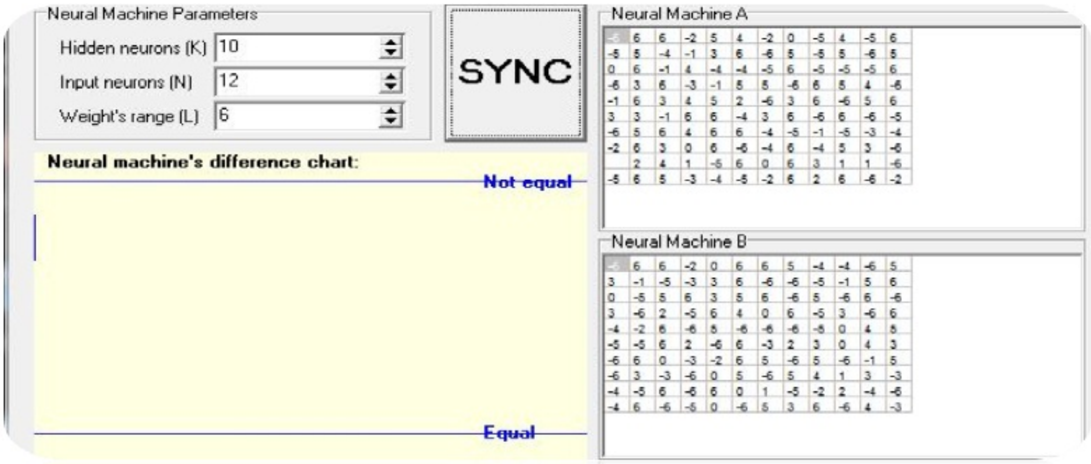


| No. of Input Neuron s(N) | No. of Activated Hidden Neurons (K) | Synaptic Weight (L) | Total No. of Iterations | Data Exchange d (Kb) |
|---|---|---|---|---|
| 5 | 15 | 3 | 624 | 48 |
| 30 | 4 | 4 | 848 | 102 |
| 25 | 5 | 3 | 241 | 30 |
| 20 | 10 | 3 | 1390 | 276 |
| 8 | 15 | 4 | 2390 | 289 |

## 6 Analysis of Results

From results obtained it is clear that the technique will achieve optimal performances. Encryption time and decryption time varies almost linearly with respect to the block size. For the algorithm presented, Chi-Square value is very high compared to some existing algorithms. A user input key has to transmit over the public channel all the way to the receiver for performing the decryption procedure. So there is a likelihood of attack at the time of key exchange. To defeat this insecure secret key generation technique a neural network based secret key generation technique has been devised. The security issue of existing algorithm can be improved by using CKE secret session key generation technique. In this case, the two partners $A$ and $B$ do not have to share a common secret but use their indistinguishable weights or output of activated hidden layer as a secret key needed for encryption. The fundamental conception of CKE based key exchange protocol focuses mostly on two key attributes of CKE. Firstly, two nodes coupled over a public channel will synchronize even though each individual network exhibits disorganized behavior. Secondly, an outside network, even if identical to the two communicating networks, will find it exceptionally difficult to synchronize with those parties, those parties are communicating over a public network. An attacker $E$ who knows all the particulars of the algorithm and records through this channel finds it thorny to synchronize with the parties, and hence to calculate the common secret key. Synchronization by mutual learning (A and B) is much quicker than learning by listening (E) [12]. For usual cryptographic systems, we can improve the safety of the protocol by increasing of the key length. In the case of CKE, we improved it by increasing the synaptic depth L of the neural networks.

For a brute force attack using K hidden neurons, $K \times N$ input neurons and boundary of weights $L$, gives $(2L + 1)KN$ possibilities. For example, the configuration



Neural Machine Parameters

Hidden neurons (K) 10

Input neurons (N) 12

Weight's range (L) 6

Neural Machine A

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| -5 | 6 | 6 | -2 | 5 | 4 | -2 | 0 | -5 | 4 | -5 | 6 |
| -5 | 5 | -4 | -1 | 3 | 6 | -6 | 5 | -5 | 5 | -6 | 5 |
| 0 | 6 | -1 | 4 | -4 | -4 | -5 | 6 | -5 | -5 | -5 | 6 |
| -6 | 3 | 6 | -3 | -1 | 5 | 5 | -6 | 6 | 5 | 4 | -6 |
| -1 | 6 | 3 | 4 | 5 | 2 | -6 | 3 | 6 | -6 | 5 | 6 |
| 3 | 3 | -1 | 6 | 6 | -4 | 3 | 6 | -6 | 6 | -6 | -5 |
| -6 | 5 | 6 | 4 | 6 | 6 | -4 | -5 | -1 | -5 | -3 | -4 |
| -2 | 6 | 3 | 0 | 6 | -6 | -4 | 6 | -4 | 5 | 3 | -6 |
| | 2 | 4 | 1 | -5 | 6 | 0 | 6 | 3 | 1 | 1 | -6 |
| -5 | 6 | 5 | -3 | -4 | -5 | -2 | 6 | 2 | 6 | -6 | -2 |

Neural Machine B

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| -5 | 6 | 6 | -2 | 0 | 6 | 6 | 5 | -4 | -4 | -6 | 5 |
| 3 | -1 | -5 | -3 | 3 | 6 | -6 | -6 | -5 | -1 | 5 | 6 |
| 0 | -5 | 5 | 6 | 3 | 5 | 6 | -6 | 5 | -6 | 6 | -6 |
| 3 | -6 | 2 | -5 | 6 | 4 | 0 | 6 | -5 | 3 | -6 | 6 |
| -4 | -2 | 6 | -6 | 5 | -6 | -6 | -6 | -5 | 0 | 4 | 5 |
| -5 | -5 | 6 | 2 | -6 | 6 | -3 | 2 | 3 | 0 | 4 | 3 |
| -6 | 6 | 0 | -3 | -2 | 6 | 5 | -6 | 5 | -6 | -1 | 5 |
| -6 | 3 | -3 | -6 | 0 | 5 | -6 | 5 | 4 | 1 | 3 | -3 |
| -4 | -5 | 6 | -6 | 6 | 0 | 1 | -5 | -2 | 2 | -4 | -6 |
| -4 | 6 | -6 | -5 | 0 | -6 | 5 | 3 | 6 | -6 | 4 | -3 |

Figure 4: CKE Key Simulation Snapshot with N=12, K=10 and L=6

$K = 3$, $L = 3$ and $N = 100$ gives us $3 \times 10253$ key possibilities, making the attack unfeasible with today's computer power. $E$ could start from all of the $(2L + 1)3N$ initial weight vectors and calculate the ones which are consistent with the input/output sequence. It has been shown, that all of these initial states move towards the same final weight vector, the key is unique. This is not true for simple perceptron the most unbeaten cryptanalysis has two supplementary ingredients first; a group of attacker is used. Second, $E$ makes extra training steps when $A$ and $B$ are quiet [1, 12, 13]. So increasing synaptic depth L of the CKE we can make our CKE safe.

## 7 Security Issue

The main difference between the partners and the attacker in CKE is that $A$ and $B$ are able to influence each other by communicating their output bits $\tau^A$ and $\tau^B$ while $E$ can only listen to these messages. Of course, $A$ and $B$ use their advantage to select suitable input vectors for adjusting the weights which finally leads to different synchronization times for partners and attackers. However, there are more effects, which show that the two-way communication between $A$ and $B$ makes attacking the CKE protocol more difficult than simple learning of examples. These confirm that the security of CKE key generation is based on the bidirectional interaction of the partners. Each partener uses a seperate, but identical pseudo random number generator. As these devices are initialized with a secret seed state shared by $A$ and $B$. They produce exactly the same sequence of input bits. Whereas attacker does not know this secret seed state. By increasing synaptic depth average synchronize time will be increased by polynomial time. But success probability of attacker will be drop exponentially Synchonization by mutual learning is much faster than learning by adopting to example generated by other network. Unidirectional learning and bidirectional synchronization. As $E$ can't influence $A$ and $B$ at the time they stop transmit due to synchrnization. Only one weight get changed where, $\sigma_i = T$. So, difficult to find weight for attacker to know the actual weight without knowing internal representation it has to guess.

## 8 Conclusion

This paper presented a novel approach for cryptanalysis of key exchange using multilayer perceptron. This technique enhances the security features of the key exchange algorithm by increasing of the synaptic depth $L$ of the CKE. Here two partners $A$ and $B$ do not have to exchange a common secret key over a public channel but use their indistinguishable weights or outputs of the activated hidden layer as a secret key needed for encryption or decryption. So likelihood of attack proposed technique is much lesser than the simple key exchange algorithm. Future scope of this technique is that this CKE model can be used in wireless communication and also in key distribution mechanism.

## Acknowledgments

The author expresses deep sense of gratitude to the Department of Science & Technology (DST) , Govt. of India, for financial assistance through INSPIRE Fellowship leading for a PhD work under which this work has been carried out, at the department of Computer Science & Engineering, University of Kalyani.

## References

[1] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning*, Cambridge University Press, Cambridge, 2001.

[2] T. Godhavari, N. R. Alainelu and R. Soundararajan, "Cryptography using neural network," in *IEEE Indicon'05*, Chennai, India, pp. 258–261, Dec. 2005.

[3] D. Hu, "A new service based computing security model with neural cryptography," in *Second Pacific-Asia Conference on Web Mining and Web-based Application (WMWA'09)*, pp. 154–156, 2009.

[4] A. Kahate, *Cryptography and Network Security*, Tata McGraw-Hill, 2006.

[5] W. Kinzel and L. Kanter, "Interacting neural networks and cryptography," in *Advances in Solid State Physics*, vol. 42, pp. 383, 2002.

[6] J. K. Mandal, A. Sarkar, "Multilayer perceptron guided key generation through mutation with recursive replacement in wireless communication," *International Journal on AdHoc Networking Systems*, vol. 2, no. 3, pp. 11-28, 2012.

[7] J. K. Mandal, A. Sarkar, "Neural weight session key based encryption for online wireless communication," in *Research and Higher Education in Computer Science and Information Technology (RHECSIT'12)*, pp. 90–95, Feb. 2012.

[8] J. K. Mandal, A. Sarkar, "An adaptive genetic key based neural encryption for online wireless communication," in *International Conference on Recent Trends In Information Systems (RETIS'11)*, pp. 62–67, Dec. 2011.

[9] J. K. Mandal, A. Sarkar, "An adaptive neural network guided secret key based encryption through recursive positional modulo-2 substitution for online wireless communication," in em International Conference on Recent Trends In Information Technology (ICRTIT'11), pp. 107–112, 2011.

[10] J. K. Mandal, A. Sarkar, "An adaptive neural network guided random block length based cryptosystem," in *2nd International Conference on Wireless Communications, Vehicular Technology, Information Theory And Aerospace & Electronic System Technology*, pp. 1–5, Feb. 2011.

[11] J. K. Mandal, A. Sarkar, "Neural network guided secret key based encryption through cascading chaining of recursive positional substitution of prime nonprime," in *International Confference on Computing and Systems (ICCS'10)*, pp. 291–297, Nov. 2010.

[12] R. Mislovaty, Y. Perchenok, I. Kanter, and W. Kinzel, "Secure key-exchange protocol with an absence of injective functions," *Physical Review E*, vol. 66, no. 6, Dec. 2002.

[13] A. Ruttor, W. Kinzel, R. Naeh, and I. Kanter, "Genetic attack on neural cryptography," *Physical Review E*, vol. 73, no. 3, pp.1–8, 2006.

[14] A. Sarkar, S. Karforma, J. K. Mandal, "Object oriented modeling of IDEA using GA based efficient key generation for E-governance security (OOMIG)," *International Journal of Distributed and Parallel Systems*, vol. 3, no. 2, pp. 171-183, Mar. 2012.

[15] A. Sarkar, J. K. Mandal, *Artificial Neural Network Guided Secured Communication Techniques: A Practical Approach*, LAP Lambert Academic Publishing, ISBN: 978-3-659-11991-0, 2012.

**Arindam Sarkar** INSPIRE FELLOW (DST, Govt. of India), MCA (VISVA BHARATI, Santiniketan, University First Class First Rank Holder), M.Tech (CSE, K.U, University First Class First Rank Holder).

**Jyotsna Kumar Mandal** M. Tech. (Computer Science, University of Calcutta), Ph.D.(Engg., Jadavpur University) in the field of Data Compression and Error Correction Techniques, Professor in Computer Science and Engineering, University of Kalyani, India. Life Member of Computer Society of India since 1992 and life member of cryptology Research Society of India. Dean Faculty of Engineering, Technology & Management, working in the field of Network Security, Steganography, Remote Sensing & GIS Application, Image Processing. 25 years of teaching and research experiences. Eight Scholars awarded Ph.D. and 8 are pursuing.

# Security Analysis of a Pairing-free Identity-based Authenticated Group Key Agreement Protocol for Imbalanced Mobile Networks

Qingfeng Cheng[1,2]
Department of Language Engineering & Luoyang University of Foreign Languages[1]
Luoyang 471003, P.R. China
Science and Technology on Information Assurance Laboratory, Beijing 100072, P.R.China[2]
(Email: qingfengc2008@sina.com)

*(Received Jan. 7, 2013; revised and accepted Aug. 27 & Nov. 10, 2013)*

## Abstract

Recently, Isalam and Biswas proposed a new group key agreement (GKA) protocol for imbalanced mobile networks. In this letter, we will first prove that Isalam and Biswas's GKA protocol cannot provide perfect forward secrecy. Then we will point out that their GKA protocol is vulnerable to ephemeral key compromise attack.

*Keywords: Ephemeral key compromise attack, group key agreement, imbalanced mobile networks, perfect forward secrecy*

## 1 Introduction

Mobile network is an imbalanced wireless network, where users have different computing capability. For assuring secure communications in mobile network, in general it needs to encrypt the messages transmitted by users, which means that users must generate shared session keys before starting communications. There are many two-party and group authenticated key agreement (AKA) protocols [1, 2, 6, 7, 8, 12] for imbalanced wireless network. However, the design of secure AKA protocols for imbalanced mobile networks is not a trivial task.

In [9], Nam et al. proposed an efficient group key agreement (GKA) protocol based on the Decisional Diffie-Hellman assumption for imbalanced mobile networks. Nam et al.'s construction was simple, and met many security properties. However, Tseng [11] pointed out that Nam et al.'s protocol still had a weakness, i.e. lack of contributory property. Further, Tseng [11] proposed a new GKA protocol with contributory property, whereas Tseng's protocol did not consider mutual authentication due to Lee et al. [5]. For achieving mutual authentication, Lee et al. [5] presented a new GKA protocol proven secure in a security model. Unfortunately, Lee et al.'s protocol is not secure due to Cheng et al. [3] and Tsai [10] respectively.

Recently, Isalam and Biswas [4] also proposed a new GKA protocol for imbalanced mobile networks, called Isalam-Biswas protocol. They claimed that their protocol met various attributes, including perfect forward secrecy and ephemeral key compromise resilience. In this letter, however, we will show that the Isalam-Biswas protocol cannot provide perfect forward secrecy. In addition, we also prove that the Isalam-Biswas protocol cannot resist ephemeral key compromise attack.

## 2 Review of Isalam-Biswas Protocol

### 2.1 System Initialization Stage

Let $k$ be a security parameter, $G$ be an additive group of prime order $q$. $P$ is a generator of group $G$. The key generation center (KGC) randomly chooses a value $s \in Z_q^*$ as the master private key and computes $P_{pub} = sP$ as its master public key. The KGC chooses two hash functions $H_0 : \{0,1\}^* \times G \longrightarrow Z_q^*$ and $H_1 : \{0,1\}^* \longrightarrow \{0,1\}^k$. The system parameters are $\{q, G, P, H_0, H_1\}$.

### 2.2 Key Extract Stage

The KGC first randomly chooses $v_i \in Z_q^*$ for each user $U_i(1 \leq i \leq n-1)$, whose identity is $ID_i \in \{0,1\}^*$. Then the KGC computes $R_i = v_iP, h_i = H_0(ID_i \parallel R_i)$ and $u_i = v_i + h_is$. Finally, the user's private key is $(u_i, R_i)$.

### 2.3 Group Key Agreement Stage

we suppose low-power user $U_i(1 \leq i \leq n-1)$ and powerful user $U_n$ wish to agree a shared group session key.

**Step 1.** Each user $U_i(1 \leq i \leq n-1)$ randomly chooses $r_i \in Z_q^*$, and computes $M_i = r_iu_iP$. Then $U_i(1 \leq$

$i \leq n-1$) computes

$$S_i = u_i(H_1(ID_i \parallel M_i) + r_i.$$

Finally, $U_i(1 \leq i \leq n-1)$ sends $\{ID_i, M_i, S_i, R_i\}$ to powerful user $U_n$.

**Step 2.** Upon receiving $\{ID_i, M_i, S_i, R_i\}$, $U_n$ checks the equations $S_iP - H_1(ID_i \parallel M_i)P_i = M_i$ for $1 \leq i \leq n-1$. If one of them fails, $U_n$ terminates the session. Otherwise, $U_n$ randomly chooses $r_n \in Z_q^*$, and computes $M_n = r_n u_n P$ and $Z_i = r_n u_n(M - M_i)(1 \leq i \leq n-1)$. Then $U_n$ sets

$$\begin{aligned} M &= M_1 + M_2 + \cdots + M_{n-1}, \\ ID &= ID_1 \parallel ID_2 \parallel \cdots \parallel ID_n, \\ Z &= Z_1 \parallel Z_2 \parallel \cdots \parallel Z_{n-1}, \end{aligned}$$

and computes

$$\begin{aligned} K &= r_n u_n M \\ &= r_n u_n (r_1 u_1 + r_2 u_2 + \cdots + r_{n-1} u_{n-1})P, \\ S_n &= u_n(H_1(ID_n \parallel Z \parallel M_n) + r_n). \end{aligned}$$

Finally, $U_n$ sends $\{ID_n, M_n, Z_1, \cdots, Z_{n-1}, S_n, R_n\}$ to each user $U_i(1 \leq i \leq n-1)$.

**Step 3.** Upon receiving $\{ID_n, M_n, Z_1, \cdots, Z_{n-1}, S_n, R_n\}$, $U_i(1 \leq i \leq n-1)$ checks the equation $S_nP - H_1(ID_n \parallel Z \parallel M_n)P_n = M_n$. If it fails, $U_i(1 \leq i \leq n-1)$ terminates the session. Otherwise, $U_i(1 \leq i \leq n-1)$ sets $ID = ID_1 \parallel ID_2 \parallel \cdots \parallel ID_n$ and computes

$$K = K_i = r_i u_i M_n + Z_i.$$

Finally, $U_i(1 \leq i \leq n-1)$ generates the group session key as follows:

$$GSK = H_1(ID \parallel Z \parallel K).$$

# 3 Analysis of Isalam-Biswas Protocol

## 3.1 Attack 1

In this subsection, we present the first attack against the Isalam-Biswas protocol. We will show that the Isalam-Biswas protocol cannot provide perfect forward secrecy.

We assume the adversary $E$ has achieved $U_1$'s private key $u_1$. Now, the adversary $E$ can first compute $u_1^{-1}$ and $H_1(ID_1 \parallel M_1)$. Then the adversary $E$ can compute $r_1$ as follows:

$$r_1 = S_1 u_1^{-1} - H_1(ID_1 \parallel M_1).$$

It means that the adversary $E$ can use the random number $r_1$ and private key $u_1$ to compute $K$ as follows:

$$K = K_1 = r_1 u_1 M_n + Z_1.$$

Clearly, the adversary $E$ now can generate the group session key $GSK = H_1(ID \parallel Z \parallel K)$ successfully, since $ID$ and $Z$ are public messages. So the Isalam-Biswas protocol cannot provide perfect forward secrecy.

## 3.2 Attack 2

In this subsection, we present our second attack, i.e. ephemeral key compromise attack, against the Isalam-Biswas protocol. In the original Isalam-Biswas protocol, the authors claimed even if all ephemeral values $(r_1, \cdots, r_n)$ were disclosed, the accepted group session key still was secure. However, we will show that the Isalam-Biswas protocol cannot resist ephemeral key compromise attack. Here, we only assume the adversary $E$ has obtained $U_1$'s ephemeral key $r_1$.

Now, the adversary $E$ can first compute $H_1(ID_1 \parallel M_1) + r_1$, and then computes $(H_1(ID_1 \parallel M_1) + r_1)^{-1}$. Finally, the adversary $E$ can compute $u_1$ as follows:

$$u_1 = S_1(H_1(ID_1 \parallel M_1) + r_1)^{-1}.$$

It means that the adversary $E$ can use the random number $r_1$ and private key $u_1$ to compute $K$ as follows:

$$K = K_1 = r_1 u_1 M_n + Z_1.$$

Clearly, the adversary $E$ now can generate the group session key $GSK = H_1(ID \parallel Z \parallel K)$ successfully, since $ID$ and $Z$ are public messages. So the Isalam-Biswas protocol cannot resist ephemeral key compromise attack.

# 4 Conclusions

In this letter, we have pointed out that Isalam et al.'s protocol is insecure against ephemeral key compromise attack. Moreover, we show that Isalam et al.'s protocol cannot provide perfect forward secrecy. For overcoming these security flaws, it needs to carefully select a secure signature scheme to improve Isalam et al.'s protocol.

# Acknowledgments

The author gratefully thanks Prof. Min-Shiang Hwang and the anonymous reviewers for their valuable comments. This study was supported by the Science Foundation of Luoyang University of Foreign Languages (No. 2011XYZ004) and Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-13-109).

# References

[1] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval, "Mutual authentication and group key agreement for low-power mobile devices," *Computer Communications*, vol. 27, no. 17, pp. 1730–1737, 2004.

[2] Y. Chang, C. Chang, and J. Yang, "An efficient password authenticated key exchange protocol for imbalanced wireless networks," *Computers Standards and Interfaces*, vol. 27, no. 3, pp. 313–322, 2005.

[3] Q. Cheng, C. Ma, and F. Wei, "Analysis and improvement of a new authenticated group key agreement in a mobile environment," *Annals of Telecommunications*, vol. 66, no. 5–6, pp. 331–337, 2011.

[4] S. Islam and G. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of Telecommunications*, vol. 67, no. 11–12, pp. 547–558, 2012.

[5] C. Lee, T. Lin, and C. Tsai, "A new authenticated group key agreement in a mobile environment," *Annals of Telecommunications*, vol. 64, no. 11–12, pp. 735–744, 2009.

[6] J. Lo, "The improvement of ysyct scheme for imbalanced wireless network," *International Journal of Network Security*, vol. 3, no. 1, pp. 39–43, 2006.

[7] J. Lo, J. Lee, M. Hwang, and Y. Chu, "An advanced password authenticated key exchange protocol for imbalanced wireless networks," *Journal of Internet Technology*, vol. 11, no. 7, pp. 997–1004, 2010.

[8] J. Nam, S. Kim, and D. Won., "A weakness in the Bresson-Chevassut-Essiari-Pointcheval's group key agreement scheme for low-power mobile devices," *IEEE Communications Letters*, vol. 9, no. 5, pp. 429–431, 2005.

[9] J. Nam, J. Lee, S. Kim, and D. Won., "DDH-based group key agreement in a mobile environment," *Journal of Systems Software*, vol. 78, no. 1, pp. 73–83, 2005.

[10] J. Tsai, "A novel authenticated group key agreement protocol for mobile environment," *Annals of Telecommunications*, vol. 66, no. 11–12, pp. 663–669, 2011.

[11] Y. Tseng, "A resource-constrained group key agreement protocol for imbalanced wireless networks," *Computer Security*, vol. 26, no. 4, pp. 331–337, 2007.

[12] H. Yeh, H. Sun, C. Yang, B. Chen, and S. Tseng, "The improvement of password authenticated key exchange scheme based on RSA for imbalanced wireless network," *IEICE Transactions on Communications*, vol. E86-B, no. 11, pp. 3278–3282, 2003.

**Qingfeng Cheng** received his B.A. degree in 2000 and M.S. degree in 2004 from National University of Defense Technology, and Ph.D. degree in 2011 from Information Engineering University. He is now an Associate Professor with the Department of Language Engineering, Luoyang University of Foreign Languages. His research interests include cryptography and information security.

# Guide for Authors
## International Journal of Network Security

IJNS will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of network security. Topics will include, but not be limited to, the following: Biometric Security, Communications and Networks Security, Cryptography, Database Security, Electronic Commerce Security, Multimedia Security, System Security, etc.

### 1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijns.femto.com.tw/.

### 2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

### 2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

### 2.2 Title page

The title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

### 2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

### 2.4 References

References should be listed alphabetically, in the same way as follows:

For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

### 2.5 Author benefits

No page charge is made.

## Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 6,000 (Taiwan). The rate is US$600.00 or NT 19,800 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Femto Technique Co., LTD." For detailed information, please refer to http://ijns.femto.com.tw or Email to ijns.publishing@gmail.com.