IJNS
International Journal of Network Security

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

Vol. 18, No. 6 (Nov. 2016)

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. 18, No. 6 (Nov. 1, 2016)

Secure Chaotic Maps-based Group Key Agreement Scheme with Privacy Preserving

Hongfeng Zhu
(Corresponding author: Hongfeng Zhu)
Software College, Shenyang Normal University
No.253, Huang He Bei Street, Huang Gu District, Shenyang 110034, P. R. China
(Email: zhuhongfeng1978@163.com)
(Received May 27, 2014; revised and accepted Oct. 27 & Dec. 24, 2014)

Abstract

Nowadays chaos theory related to cryptography has been addressed widely, so there is an intuitive connection between group key agreement and chaotic maps. Such a connector may lead to a novel way to construct authenticated and efficient group key agreement protocols. Many chaotic maps based two-party/three-party password authenticated key agreement (2PAKA/3PAKA) schemes have been proposed. However, to the best of our knowledge, no chaotic maps based group (N-party) key agreement protocol without using a timestamp and password has been proposed yet. In this paper, we propose the first chaotic maps-based group authentication key agreement protocol. The proposed protocol is based on chaotic maps to create a kind of signcryption method to transmit authenticated information and make the calculated consumption and communicating round restrict to an acceptable bound. At the same time our proposed protocol can achieve members' revocation or join easily, which not only refrains from consuming modular exponential computing and scalar multiplication on an elliptic curve, but is also robust to resist various attacks and achieves perfect forward secrecy with privacy preserving.

Keywords: Authentication, chaotic maps, group key, random oracle model

1 Introduction

In the network information era, it is important to structure group key agreement schemes which are designed to provide a set of players, and communicating over a public network with a session key to be used to implement secure multicast sessions, e.g., video conferencing, collaborative computation, file sharing via internet, secure group chat, group purchase of encrypted content and so on.

With the rapid development of chaos theory related to cryptography [3, 4, 15, 16, 18, 34], many key agreement protocols using a chaotic map have been studied widely. These protocols using a chaotic map can mainly be divided into three directions: two-party authenticated key agreement protocols [2, 9, 10, 11, 12, 13, 14, 24, 25, 26, 27, 28, 31, 32, 33, 37, 39], three-party authenticated key agreement protocols [8, 19, 20, 29, 30, 36, 38, 40], and N-party authenticated key agreement protocols. Furthermore, we can classify the literatures [2, 8, 9, 10, 11, 12, 13, 14, 19, 20, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 36, 37, 38, 39, 40] based on their respective features in detail, such as password-based, using smart card, timestamp, anonymity and other security attributes. From the macroscopic point of view, these literatures have two main traits: On the one hand, along with some new protocols putting forward, then some flaws will be found over a period of time, such as the flaws in the literatures [11, 25, 32] are found by the literatures [2, 12, 14]. On the other hand, the evolution of the key agreement protocols using a chaotic map shows putting in new secure attributes and improving the efficiency, for example the literatures [13, 28, 33, 37]. In recent years, the three-party password-authenticated key agreement protocol using modular exponentiation or scalar multiplication on an elliptic curve has been addressed widely [30, 38]. However, these schemes need heavy computation costs and even most recent the research is still remaining on three-party authenticated key agreement protocol [36].

To the best of our knowledge, no N-party authenticated key agreement protocol based on chaotic maps has been proposed, yet. To design group authentication key agreement protocols in chaotic map setting is difficult but is very useful in many application environments. The difficult of the setting is when the number of participants increasing, and how to keep computing and communication increasing linearly or constantly. So it is quite natural to utilize N-party authenticated key agreement literature that related to cryptography. The first work in this area is by Bresson et al. [21]. As already mentioned, their proposed scheme is secure in both the random oracle model and the ideal cipher model. Next Lee presents a password-based group key protocol [5] which is not authenticated

because there is no way to convince a user that the message that he receives is indeed coming from the intended participant. Recently there are three literatures about password-based group key scheme [1, 7, 22, 42] and Abdalla et al. [1] points out the literature [7] which is subjected to an off-line dictionary attack, however their efficiency is unsatisfactory.

In this paper, we put forward a new simple and efficient N-party authenticated key agreement protocol based on chaotic maps. We present our contributions below:

1) Communication round: Our proposed protocol is efficient from communication point of view as it requires only 2 rounds and uses Chebyshev chaotic maps and symmetric key encryption instead of signature for message authentication in the round 1. And in the round 2, we mainly use hash function and operations to authenticated each other and compute the group session key. These methods reduce the bandwidth of the messages sent and make the protocol faster.

2) Computation: Our protocol is based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve.

3) Security: The protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc.

4) Functionality: It allows N ($N \geq 2$) users establish a secure session key over an insecure communication channel with the help of public key system with chaotic maps. The proposed protocol has provided the case of a member revocation or a new member join. Furthermore the protocol also has achieved some well-known properties, such as perfect forward secrecy, no timestamp, and execution efficiency.

The rest of the paper is organized as follows: We outline preliminaries in Section 2. Next, A Chebyshev chaotic maps-based N-party authenticated key agreement protocol is described in Section 3. Then, the security analysis and efficiency analysis are given in Section 4. This paper is finally concluded in Section 5.

2 Preliminaries

Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial. $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = \cos(n \arccos(x))$ Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n is defined using the following recurrent relation [29]:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \quad (1)$$

where $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$. The first few Chebyshev polynomials are:

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ &\vdots \quad\quad \vdots \end{aligned}$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{r \cdot s}(x). \quad (2)$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition:

$$T_r(T_s(x)) = T_s(T_r(x)). \quad (3)$$

In order to enhance the security, Zhang [41] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. In our proposed protocol, we utilize the enhanced Chebyshev polynomials:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(\mathrm{mod} N) \quad (4)$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and N is a large prime number. Obviously,

$$T_{r \cdot s}(x) = T_r(T_s(x)) = T_s(T_r(x)). \quad (5)$$

Definition 1. *Semi-group property of Chebyshev polynomials:*

$$\begin{aligned} T_r(T_s(x)) &= \cos(r\cos^{-1}(s\cos^{-1}(x))) \\ &= \cos(rs\cos^{-1}(x)) = T_{sr}(x) \\ &= T_s(T_r(x)). \end{aligned}$$

Definition 2. *Given x and y, it is intractable to find the integer s, such that $T_s(x) = y$. It is called the Chaotic Maps-Based Discrete Logarithm problem (CMBDLP).*

Definition 3. *Given x, $T_r(x)$, and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie-Hellman problem (CMBDHP).*

3 Group Key Agreement from Chaotic Maps

We now consider the generic construction for a two-round group key agreement from Chaotic Maps. All group participants $U_1, U_2, ..., U_n$ are organized in an ordered chain and U_{i+1} is the successor of U_i. The temporary two-party symmetric session key computed in a parallel algorithm based on Chaotic Maps-Based Diffie-Hellman problem is used as the shared secret between the participant U_i and its successor $U_{i+1}, i = 1, ..., n$. The structure of the kind of group key agreement from Chaotic Maps is illustrated in Figure 1 which includes the following two rounds.

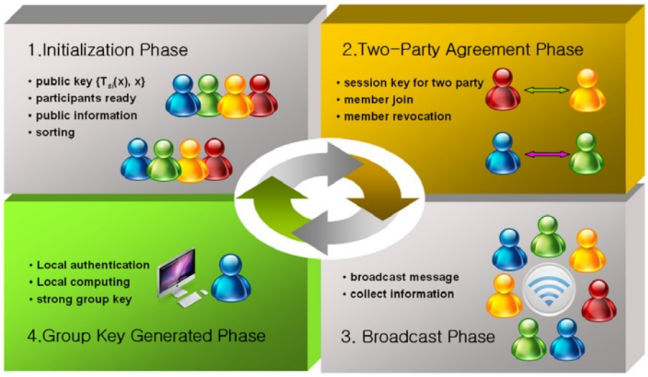


Figure 1: Structure of the PGKA phases (The phases are presented clockwise)

3.1 Setup Phase

In this phase, any user U_i has its identity ID_i, and public key $(x, T_{S_i}(x))$ and a secret key S_i based on Chebyshev chaotic maps, a chaotic maps-based one-way hash function $h(\cdot)$ [35], and a pair of secure symmetric encryption/decryption functions $E_K()/D_K()$ with key K. The concrete notation used hereafter is shown in Table 1.

3.2 Authentication and Two-party Agreement Phase

Let $U = \{U_1, U_2, ..., U_n\}$ be the set of protocol participants. All the participants $U_1, U_2, ..., U_n$ run the following process. This process is presented in Figure 2.

Remark 1. *In order to put emphasis on describing the proposed protocol, we assume that all ID information has been arranged.*

Step 1. User U_i selects a random number r_i and computes

$$\begin{aligned} K_{i,i+1} &= T_{r_i} T_{S_{i+1}}(x), \\ C_i &= E_{K_{i,i+1}}(ID_i||ID_{i+1}||T_{r_i}(x)) \\ MAC_i &= H(ID_i||ID_{i+1}||C_i||H(K_{i,i+1})||T_{r_i}(x)), \end{aligned}$$

and sends messages $\{C_i, T_{r_i}(x), MAC_i\}$ to user U_{i+1}.

Step 2. After receiving the messages $\{C_i, T_{r_i}(x), MAC_i\}$, user U_{i+1} firstly computes $T_{S_{i+1}} T_{r_i}(x) = K_{i+1,i}$ to extract C_i to get ID information. Then user U_{i+1} verifies MAC_0 through computing

$$H(ID_i||ID_{i+1}||C_i||H(K_{i+1,i})||T_{r_i}(x)).$$

If $H(ID_i||ID_{i+1}||C_i||H(K_{i+1,i})||T_{r_i}(x)) = MAC_i$ holds, then U_{i+1} selects a random number r_{i+1} and compute

$$\begin{aligned} K_{i+1,i} &= T_{r_{i+1}} T_{S_i}(x) \\ SK &= T_{r_{i+1}} T_{r_i}(x), \\ C_{i+1} &= E_{K_{i+1,i}}(ID_i||ID_{i+1}||T_{r_{i+1}}(x)), \\ MAC_{i+1} &= H(ID_i||ID_{i+1}||C_{i+1}||T_{r_{i+1}}(x) \\ &\quad ||H(K_{i+1,i})||SK). \end{aligned}$$

Finally user U_{i+1} sends messages $\{C_{i+1}, T_{r_{i+1}}(x), MAC_{i+1}\}$ to user U_i.

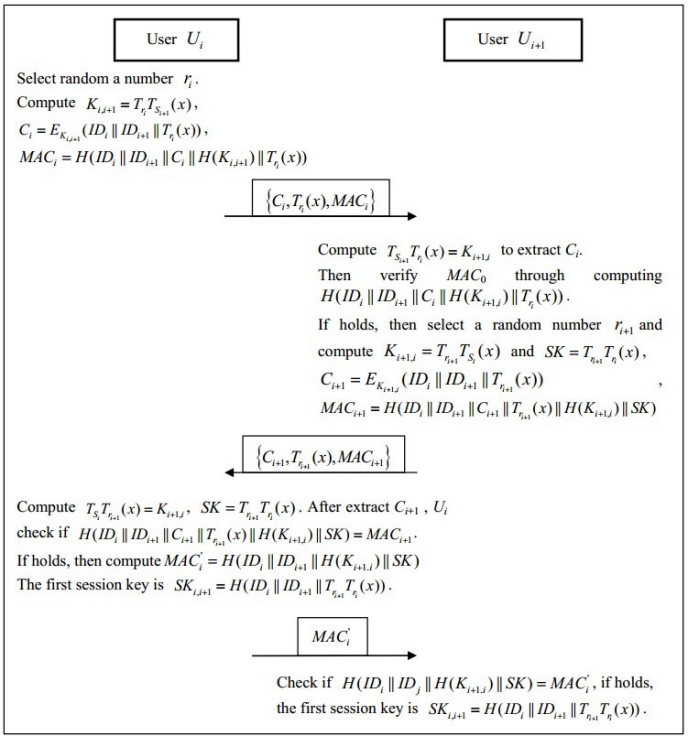


Figure 2: Two-party agreement phase

Step 3. After receiving the messages $\{C_{i+1}, T_{r_{i+1}}(x), MAC_{i+1}\}$, user U_i uses S_i and r_i to compute $T_{S_i} T_{r_{i+1}}(x) = K_{i+1,i}$ and $SK = T_{r_{i+1}} T_{r_i}(x)$. User U_i uses $K_{i+1,i}$ to extract C_{i+1} and computes $H(ID_i||ID_{i+1}||C_{i+1}||T_{r_{i+1}}(x)||H(K_{i+1,i})||SK)$ and then checks if it equals MAC_{i+1}.

If not, user U_i terminates it. Otherwise, user U_i computes $MAC_i' = H(ID_i||ID_{i+1}||H(K_{i+1.i})||SK)$ and $SK_{i,i+1} = H(ID_i||ID_{i+1}||T_{r_{i+1}} T_{r_i}(x))$. User U_i sends MAC_i' to user U_{i+1}, and at the same time takes $SK_{i,i+1} = H(ID_i||ID_{i+1}||T_{r_{i+1}} T_{r_i}(x))$ as the session key.

Step 4. Upon receiving MAC_i', user U_{i+1} computes $H(ID_i||ID_{i+1}||H(K_{i+1.i})||SK)$ and checks if it equals MAC_i'. If not, user U_{i+1} terminates it. Otherwise, user U_{i+1} uses $SK_{i,i+1} = H(ID_i||ID_{i+1}||T_{r_{i+1}} T_{r_i}(x))$ as the session key.

Table 1: Notations

Symbols	Definition
U_i, ID_i	The Participant i and its identity information;
U	Set of protocol participants;
$(x, T_{S_i}(x)$	Public key based on Chebyshev chaotic maps;
S_i	Secret key based on Chebyshev chaotic maps;
$E_K(\cdot)/D_K(\cdot)$	A pair of secure symmetric encryption/decryption functions with the key K;
r_i	Random nonce chosen by each U_i;
$\oplus$	A bitwise Xor operator;
$\|\|$	Two adjacent messages are concatenated;
H	A chaotic maps-based one-way hash function.

The phase can be simultaneous and parallel. Finally, each participant has two two-party agreement keys ($SK_{i,i+1}$ and $SK_{i-1,i}$) with its sucessor and predecessor (U_1 computes $SK_{1,2}$ and $SK_{n,1}$).

3.3 Broadcast and Group Key Agreement Generated Phase

This process is presented in Figure 3.

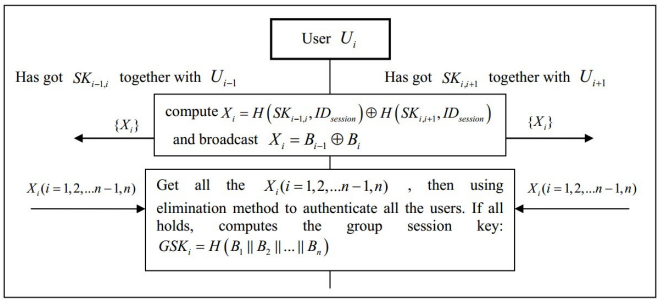


Figure 3: Group Key Agreement Generated Phase

The participants $U_i, i = 2, ..., n$, compute and broadcast X_i, where $X_i = B_{i-1} \oplus B_i = H(SK_{i-1,i}, ID_{session}) \oplus H(SK_{i,i+1}, ID_{session})$. Note that the first participant U_1 computes and broadcasts $X_1 = H(SK_{n,1}, ID_{session}) \oplus H(SK_{1,2}, ID_{session})$. Here $ID_{session}$ is the public ephemeral information that consists of participants' identities and a nonce, aiming to make the protocol secure against known-key attacks. To sum it up, we can see Table 2.

Finally, with secret $SK_{i-1,i}$ and $SK_{i,i+1}$ the participant $U_i\,(i = 1, ..., n)$ computes B_i and further get all $B_j\,(j = 1, ..., n)$ using continuous XOR method. Then, the participant $U_i, (i = 1, ..., n)$ compares B_{i-1} and$H(SK_{i-1,i}, ID_{session})$ locally. Furthermore, each participant $U_i (i = 1, ..., n)$ verifies if $X_1 \oplus X_2 \oplus X_3 \oplus ... \oplus X_{n-1} \oplus X_n = 0$ holds and all participants will continue to compute the group key. If not, output an error symbol $\perp$ and abort. After all participants accomplish the verifying, they compute the group session key $GSK_i = H(B_1||B_2||...||B_n)$. Obviously, $GSK_1 = GSK_2 = ... = GSK_n$. This will be the common strong group session key agreed by all participants.

3.4 A Member Revocation or a New Member Join Phase

A Member Revocation: Assume that a participant leaves the group. Then group members change the group size into $(n-1)$. The U_{x-1} participants U_{x-1} and U_{x+1} respectively remove the shared values $SK_{x-1,x}$ and $SK_{x,x+1}$ with U_x. The participant U_{x+1} becomes the new successor of participant U_{x-1}. Aiming to update group key, the participant U_{x-1} needs to send new message C_{x-1} to its new successor U_{x+1} and U_{x+1} needs to send new message C'_{x+1} to its new predecessor U_{x-1}. Then,the participant U_{x+1} verifies the validity of the message $\{C_{x-1}, T_{r_{x-1}}(x), MAC_{x-1}\}$ and computes the secret $SK_{x-1,x+1}$ which is a new shared secret between U_{x-1} and U_{x+1}. Each party U_j that follows U_x changes their index to $(j-1)$. Then, recomputed Section 3.3, all the $(n-1)$ participants implement the above protocol to get a new group session key.

A New Member Join: Assume that a new entity joins the group of which size is n. Then, the new participant U_{n+1}, becomes the successor of participant U_n and the participant U_1 becomes the successor of participant U_{n+1}.

The participant U_n sends message $\{C_n, T_{r_n}(x), MAC_n\}$ according to ID_n and ID_{n+1} to its new successor U_{n+1} while U_{n+1} sends message $\{C_{n+1}, T_{r_{n+1}}(x), MAC_{n+1}\}$ to U_{n+1} based on ID_nandID_{n+1}.

From the message C_n and C'_{n+1}, the new participant U_{n+1} verifies the validity of the message and computes the secret $SK_{n,n+1}$ which is the new shared secret between U_n and its new successor U_{n+1}. At the same time, the first participant U_1 updates its secret with $SK_{n+1,1}$ in Figure 4. Then, recomputed Section 3.3, the participants in the group implement the above protocol to get a new group session key.

Table 2: The value of B_i

Notations	B_1	B_2	$\cdots$	B_i	$\cdots$
Value	$H(SK_{1,2}, ID_{Session})$	$H(SK_{2,3}, ID_{Session})$	$\cdots$	$H(SK_{i,i+1}, ID_{Session})$	$\cdots$

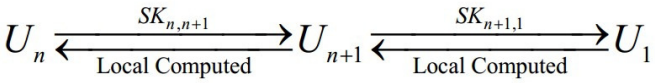


Figure 4: A new member join case

Remark 2. *The proposed protocol, when member revoke and join, the computation and communication complexity is increasing with N linearly.*

4 Security Consideration and Efficiency Analysis

Assume there are three secure components, including the two problems CMBDLP and CMBDHP cannot be solved in polynomial-time, a secure chaotic maps-based one-way hash function, and a secure symmetric encryption. Assume that the adversary has full control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages. However, the adversary could neither get the temporary values r_i chosen in the local machine nor guess ID_i correctly at the same time.

In this section, we classify the functions of group authentication key agreement scheme based on chaotic maps into two types, auxiliary function and essential function. We also prove that our proposed scheme achieves the security and efficiency goals.

4.1 Auxiliary Function

Privacy Preserving
In our protocol, the users' sensitive information such as identities is private to both the participants and the adversaries. During the whole scheme, the privacy is protected by the one-way hash function and symmetric encryption with chaotic maps-based for transferring over insecure channel and cannot be retrieved from the transmission messages. The user's identity is always combined with a nonce as $E_{K_{i,i+1}}(ID_i||ID_{i+1}||T_{r_i}(x))$ transmitting to the next participant.

Natural Resistance
Our protocol is based on public key system with chaotic maps without smart card or password, so its naturally resists many attacks, such as SEG attack [23], Password guessing attack, Stolen-verifier attack and so on.

No Clock Synchronization
The proposed protocol solves the clock synchronization problem with no timestamp mechanism. Instead, we introduce fresh random number r_i and r_{i+1} to provide the challenge response security mechanism so that replay attack cannot threaten the proposed scheme while no clock synchronization is needed.

4.2 Essential Function

Mutual Authentication, Group Authentication and Key Agreement
The proposed scheme allows the participant U_{i+1} to authenticate the participant U_i by checking whether $H(ID_i||ID_{i+1}||C_i||H(K_{i+1,i})||T_{r_i}(x)) \stackrel{?}{=} MAC_i$. Furthermore only owning the secret key S_{i+1} can extract C_i to get the secret message to verify the receiving message. About group authentication phase, each participant $U_i(i = 1, ..., n)$ verifies if $X_1 \oplus X_2 \oplus X_3 \oplus ... \oplus X_{n-1} \oplus X_n = 0$ holds and all participants will continue to compute the group key. If not, output an error symbol $\perp$ and abort.

Resist Well-known Attacks

1) Impersonation Attack/Man-in-the-Middle Attack
An adversary cannot impersonate the user U_i to cheat the participant, because it is not able to get the secret key of the user U_i and afterwards cannot extract C'_{i+1} to compute two-party session key. From the above analysis, we can know that an adversary is unable to achieve success by impersonating and replaying. On the other hand, because $\{C_i, T_{r_i}(x), MAC_i\}$, $\{C_{i+1}, T_{r_{i+1}}(x), MAC_{i+1}\}$ and $X_i, 1 \leq i \leq n$ contain the users' identities, a man-in-the-middle attack cannot succeed.
2) Replay Attack
An adversary cannot start a replay attack against our scheme because of the freshness of r_i in each session. If $T_{r_i}(x)$ has appeared before or the status shows in process, the participant U_{i+1} rejects the session request. If the adversary wants to launch the replay attack successfully, it must compute and modify $T_{r_i}(x)$ and C_i correctly which is impossible.
3) Known-key Security
Since two-party session key $SK_{i,i+1} = H(ID_i||ID_{i+1}||T_{r_i}T_{r_{i+1}}(x))$ is depended on the random nonces r_i and r_{i+1}, and the generation

Table 3: Descriptions the model of Canetti and Krawczyk

Symbols	Definition
parties $P_1, \cdots, P_n$	Modelled by probabilistic Turing machines.
Adversary *wedge*	A probabilistic Turing machine which controls all communication, with the exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once.
Send query	The adversary can control over Parties' outgoing messages via the **Send** query. Parties can be activated by the adversary launching **Send** queries.
Two sessions matching	If the outgoing messages of one are the incoming messages of the other.

of nonces is independent in all sessions, an adversary cannot compute the previous and the future session keys when he knows one session key. About the group session key $GSK_i = H(B_1||B_2||...||B_n)$ which based on all random nonces $r_i, 1 \leq i \leq n$, an adversary cannot compute the previous and the future group session keys when he knows one group session key.

4) Perfect Forward Secrecy
In the proposed scheme, the session key $SK_{i,i+1} = H(ID_i||ID_{i+1}||T_{r_i}T_{r_{i+1}}(x))$ is related with r_i and r_{i+1}, which were chosen by user U_i and user U_{i+1}, respectively. Because of the intractability of the CMBDLP and CMBDHP problem, an adversary cannot compute the previously established session keys. About the group session key $GSK_i = H(B_1||B_2||...||B_n)$ which based on all random nonces $r_i, 1 \leq i \leq n$, an adversary cannot compute the previously established group session keys yet.

5) Key Compromise Impersonation Attacks (KCI Attacks)
Informally, an adversary is said to impersonate a party B to another party A if B is honest and the protocol instance at A accepts the session with B as one of the session peers but there exists no such partnered instance at B [17]. In a successful KCI attack, an adversary with the knowledge of the long-term private key of a party A can impersonate B to A. We assume that an adversary can know U_1 and $U_3^{'}$s secret keys S_1 and S_3, then he can impersonate U_2 to cheat U_1 and U_3, and $U_4...U_n$, and to get the group session key $GSK_i = H(B_1||B_2||...||B_n)$. But above-mentioned process will not achieve and the attack course terminates at the beginning. Because an adversary cannot own the U_2's secret key S_2, and he cannot pass validation of U_3: An adversary do not possess $U_2^{'}$s secret key S_2, so he cannot compute $T_{S_i}T_{r_{i+1}}(x) = K_{i+1,i}$, and then he cannot compute the $MAC_i^{'} = H(ID_i||ID_{i+1}||H(K_{i+1.i})||SK)$, finally U_3 will check if $H(ID_i||ID_j||H(K_{i+1.i})||SK) = MAC_i^{'}$. If not, user U_3 terminates it. The key compromise impersonation attacks will fail.

4.3 The Provable Security of Our Scheme

We recall the definition of session-key security in the authenticated-links adversarial model of Canetti and Krawczyk [6]. The basic descriptions are shown in Table 3.

We allow the adversary access to the queries **SessionStateReveal, SessionKeyReveal**, and **Corrupt**.

1) SessionStateReveal(s): This query allows the adversary to obtain the contents of the session state, including any secret information. s means no further output.

2) SessionKeyReveal(s): This query enables the adversary to obtain the session key for the specified session s, so long as s holds a session key.

3) Corrupt(Pi): This query allows the adversary to take over the party P_i, including long-lived keys and any session-specific information in $P_i^{'}$s memory. A corrupted party produces no further output.

4) Test(s): This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session s. A bit b is then picked randomly. If $b = 0$, the test oracle reveals the session key, and if $b = 1$, it generates a random value in the key space. The adversary can then continue to issue queries as desired, with the exception that it cannot expose Λ the test session. At any point, the adversary can try to guess b. Let $GoodGuess^{\Lambda}(k)$ be the event that the adversary Λ correctly guesses b, and we define the advantage of adversary Λ as $Advantage^{\Lambda}(k) = \max\{0, |\Pr[GoodGuess^{\Lambda}(k)] - \frac{1}{2}|\}$, where k is a security parameter.

A session s is locally exposed with P_i: If the adversary has issued **SessionStateReveal(s)**, **SessionKeyReveal(s)**, **Corrupt(Pi)** before s is expired.

Table 4: Security of our proposed protocol

Privacy preserving	Natural resistance	No. clock synchronization	Mutual and group authentication	Impersonation
Provided	Provided	Provided	Provided	Provided
Man in the Middle Attack	Replay Attack	Known Key Security	Perfect Forward Secrecy	Key Compromise Impersonation
Provided	Provided	Provided	Provided	Provided

Table 5: Efficiency of our proposed protocol for one participant

Hash	XOR	Symmetric En/decryption	Modular Multiplication	Modular Exponent	Elliptic Curve Multiplication	Elliptic Curve Addition	Chebyshev Polynomial	Round Number
9	n	4	0	0	0	0	6	2

Definition 4. *A key exchange protocol Π_1 in security parameter k is said to be session-key secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ, is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary Λ with a non-negligible advantage ε in standard model. We claim that Algorithm 1 forms a polynomial- time distinguisher for CMBDHP having non-negligible advantage.*

1) If two uncorrupted parties have completed matching sessions, these sessions produce the same key as output;

2) $Advantage^{\Lambda}(k)$ is negligible.

Algorithm 1 CMBDHP distinguisher

Input : $H, E_K()/D_K(), (x, T_{k_A}(x)), (x, T_{k_B}(x))$

1: $r \xleftarrow{R} \{1,...,k\}$, where k is an upper bound on the number of sessions activated by Λ in any interaction.
2: Invoke Λ and simulate the protocol to Λ, except for the $r-th$ activated protocol session.
3: For the $r-th$ session, let Alice send $\{i, T_{R_A}(x), ID_A, ID_B, C_1\}$ to Bob, and let Bob send $\{i, T_{R_B}(x), ID_A, ID_B, C_2\}$ to Alice, where i is the session identifier. Both Alice and Bob can compute the session key $SK = H(T_{R_A} T_{R_B}(x))$ locally after authenticating each other by one-round messages and public information.
4: **if** the $r-th$ session is chosen by Λ as the test session **then**
5: Provide Λ as the answer to the test query.
6: $d \leftarrow \Lambda'$s output.
7: **else**
8: $d \xleftarrow{R} \{0,1\}$.
9: **end if**

Output: d

Theorem 1. *Under the CMBDHP assumption, using the Algorithm 1 to compute session key is session-key secure in the adversarial model of Canetti and Krawczyk [6].*

Proof. The proof is based on the proof given by [6, 26]. There are two uncorrupted parties in matching sessions output the same session key, and thus the first part of **Probability analysis.** It is clear that Algorithm 1 runs in polynomial time and has non-negligible advantage. There are two cases where the r-th session is chosen by Λ as the test session:

1) If the r-th session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the CMBDHP is 0.

2) If the r-th session is the test session, then Λ will succeed with advantage ε, since the simulated protocol provided to Λ is indistinguishable from the real protocol. The latter case occurs with probability $1/k$, so the overall advantage of the CMBDHP distinguisher is ε/k, which is non-negligible.

□

4.4 Practical in Pervasive and Ubiquitous Computing Environment

Compared to RSA and ECC, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. In our proposed protocol, no time-consuming modular exponentiation and scalar multiplication on elliptic curves are needed. However, Xiao et al. [34] and Wang [29] proposed several methods to solve the Chebyshev polynomial computation problem. In addition for getting the group key agreement, our proposed protocol uses hash function and $\oplus$ operations, and both of them are all high efficient algorithm.

To the best of our knowledge, no N-party authenticated key agreement protocol based on chaotic maps has been proposed, so there are no literatures to contrast and we sum up our proposed protocol as show in Table 4 (Security) and Table 5 (Efficiency). Furthermore the case of members revocation or new members join also have provided in the paper.

5 Conclusions

We put forward the first N-party authenticated key agreement protocol based on chaotic maps, symmetric key encryption, hash function and $\oplus$ operations which are all

better algorithm than RSA and ECC and so on. From the Table 5, we can see easily that ours protocol computing and communication increasing constantly along with the number of participants N, and only XOR operation increasing linearly with the number of participants N. Security of our proposed protocol is also satisfactory from the Table 4. Next we will extend the proposed protocol to high level security attributes such as fairness or entanglement and so on.

References

[1] M. Abdalla, E. Bresson E, Chevassut O, "Password-based group key exchange in a constant number of rounds," in *Proceedings of Public Key Cryptography (PKC'06)*, pp. 427–442, 2006.

[2] G. Alvarez, "Security problems with a chaos-based deniable authentication scheme," *Chaos, Solitons and Fractals*, vol. 26, no. 1, pp.7–11, 2005.

[3] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1, pp. 50–54, 1998.

[4] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, A. Akhavan, "A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps," *Physics Letters*, vol. 366, pp. 391–396, 2007.

[5] E. Bresson, O. Chevassut and D. Pointcheval, "Group Diffie-Hellman key exchange secure against dictionary attack," in *Advances in Cryptography (Asiacrypt'02)*, LNCS 2501, pp. 497–514, Springer, 2002.

[6] R. Canetti, and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptography (EUROCRYPT'01)*, LNCS 2045, pp. 453–474, Springer, 2001.

[7] R. Dutta, R. Barua, "Password-based encrypted group key agreement," *International Journal of Network Security*, vol. 3, no. 1, PP. 23–34, July 2006.

[8] T. H. Feng, C. H. Ling, and M. S. Hwang, "Cryptanalysis of Tan's Improvement on a Password Authentication Scheme for Multi-server Environments," *International Journal of Network Security*, vol. 16, no. 4, pp. 318–321, 2014.

[9] P. Gong, P. Li, W. Shi, "A secure chaotic maps-based key agreement protocol without using smart cards," *Nonlinear Dynamics*, vol. 70, pp. 2401–2406, 2012.

[10] C. Guo, C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1433–1440, 2012.

[11] X. Guo, J. Zhang, "Secure group key agreement protocol based on chaotic Hash," *Information Sciences*, vol. 180, no. 20, pp. 4069–4074, 2010.

[12] S. Han, "Security of a key agreement protocol based on chaotic maps," *Chaos Solitons Fractals*, pp. 764–768, 2008.

[13] S. Han, E. Chang, "Chaotic map based key agreement without clock synchronization," *Chaos Solitons Fractals*, vol. 39, pp. 1283–1289, 2009.

[14] D. He, "Cryptanalysis of a key agreement protocol based on chaotic hash," *International Journal of Electronic Security and Digital Forensics*, vol. 5, no. 3, pp.172–177, 2013.

[15] I. Hussain, T. Shah, M. Gondal, "A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 1791–1794, 2012.

[16] I. Hussain, T. Shah, M. Gondal, and H. Mahmood, "An efficient approach for the construction of LFT S-boxes using chaotic logistic map," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 133–140, 2013.

[17] J. Katz, J. S. Shin, "Modelling insider attacks on group key-exchange protocols," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 180–189, 2005.

[18] M. Khan, T. Shah, H. Mahmood, M. Gondal, "An efficient method for the construction of block cipher with multi-chaotic systems," *Nonlinear Dynamics*, vol. 71, pp. 489–492, 2013.

[19] H. Lai, J. Xiao, L. Li, Y. Yang, "Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol," *Mathematical Problems in Engineering*, vol. 2012, Article ID 454823, 2012.

[20] C. Lee, C. Li, C. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, pp. 125–132, 2013.

[21] S. M. Lee, J. Y. Hwang and D. H. Lee, "Efficient password-based group key exchange," in *Proceedings of TrustBus*, pp. 191–199, 2004.

[22] H. Li, C. K. Wu, J. Sun, "A general compiler for password-authenticated group key exchange protocol," *Information Processing Letters*, vol. 110, pp. 160–167, 2010.

[23] T. H. Liu, Q. Wang, H. F. Zhu, "A multi-function password mutual authentication key agreement scheme with privacy preserving," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 165–178, 2014.

[24] Y. Niu, X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communication Nonlinear*, vol. 16, no. 4, pp.1986–1992, 2011.

[25] F. Özkaynak, S. Yavuz, "Designing chaotic S-boxes based on time-delay chaotic system," *Nonlinear Dynamics*, vol. 74, no. 3, pp. 551–557, 2013

[26] A. Stolbunov, "Reductionist security arguments for public-key cryptographic schemes based on group action," in *The Norwegian Information Security Conference (NISK'09)*, pp. 97–109, 2009.

[27] Z. Tan, "A chaotic maps-based authenticated key agreement protocol with strong anonymity," *Nonlinear Dynamics*, vol. 72, pp. 311–320, 2013.

[28] H. Tseng, R. Jan, W. Yang, "A chaotic maps-based key agreement protocol that preserves user anonymity," *International Conference on Communications*, pp. 1–6, 2009.

[29] X. Wang, J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 4052–4057, 2010.

[30] S. Wu, K. Chen, Q. Pu, Y. Zhu, "Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme," *International Journal of Communication Systems*, vol. 26, no. 5, pp. 674–686, 2013.

[31] T. Xiang, K. Wong, X. Liao, "On the security of a novel key agreement protocol based on chaotic maps," *Chaos Solitons Fractals*, vol. 40, pp. 672–675, 2009.

[32] D. Xiao, X. Liao, S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Science*, vol. 177, pp. 1136–1142, 2007.

[33] D. Xiao, X. Liao, S. Deng, "Using time-stamp to improve the security of a chaotic maps-based key agreement protocol," *Information Sciences*, vol. 178, no. 6, pp. 1598–1602, 2008.

[34] D. Xiao, X. Liao, K. Wong, "An efficient entire chaos-based scheme for deniable authentication," *Chaos Solitons and Fractals*, vol. 23, no. 4, pp. 1327–1331, 2005.

[35] D. Xiao, F. Shih, X. Liao, "A chaos-based hash function with both modification detection and localization capabilities," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 2254–2261, 2010.

[36] Q. Xie, J. M. Zhao, X. Y. Yu, "Chaotic maps-based three party password-authenticated key agreement scheme," *Nonlinear Dynamics*, vol. 74, pp. 1021–1027, 2013.

[37] K. Xue, P. Hong, "Security improvement on an anonymous key agreement protocol based on chaotic maps," *Communication Nonlinear*, vol. 17, pp. 2969–2977, 2012.

[38] J. Yang, T. Cao, "Provably secure three-party password authenticated key exchange protocol in the standard model.," *Journal of Systems and Software*, vol. 85, pp. 340–350, 2012.

[39] E. Yoon, "Efficiency and security problems of anonymous key agreement protocol based on chaotic maps," *Communication Nonlinear*, vol. 17, pp. 2735–2740, 2012.

[40] E. Yoon, I. Jeon, "An efficient and secure Diffie-Hellman key agreement protocol based on Chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, pp. 2383–2389, 2011.

[41] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[42] M. H. Zheng, H. H. Zhou, J. Li, G. H. Cui, " Efficient and provably secure password-based group key agreement protocol," *Computer Standards and Interfaces*, vol. 31, no. 5, pp. 948–953, 2009.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Dr. Zhu had published more than 50 international journal papers (SCI or EI journals) on the above research fields.

An Advanced Anonymous and Biometrics-based Multi-server Authentication Scheme Using Smart Cards

Chin-Chen Chang[1], Wei-Yuan Hsueh[2], and Ting-Fang Cheng[1]
(Corresponding author: Chin-Chen Chang)
Department of Information Engineering and Computer Science, Feng Chia University[1]
No. 100, Wenhwa Rd., Seatwen, Taichung, Taiwan, 40724, R.O.C.
Department of Computer Science and Information Engineering, National Chung Cheng University[2]
No.168, Sec. 1, University Rd., Min-Hsiung Township, Chiayi, Taiwan, 62102, R.O.C.
(Email: alan3c@gmail.com)
(Received July 8, 2014; revised and accepted Jan. 16, 2015)

Abstract

We find that Chuang and Chen's biometrics-based multi-server authentication scheme is unable to resist against stolen smart card and forgery attacks; in addition, their scheme has weak biometrics detection and privacy preservation problems. Thus, in this paper, we propose an advanced biometrics-based authentication scheme for a multi-server environment with higher security and efficiency. Our scheme not only resists potential attacks but satisfies various additional requirements as well. Compared with related biometrics-based schemes, our scheme not only ensures security but also has lower computation cost. In particular, our scheme overcomes the false negative problem in biometrics detection.

Keywords: Authentication, biometrics, false negative, key agreement, multi-server

1 Introduction

Due to the development of the Internet and the convenience it provides, network identity authentication has become an important security issue that can authenticate the validity of any two communication parties on the Internet. Lamport's design [11] in 1981 was the first remote user authentication scheme in an insecure environment; however, in this scheme, the remote server has to store a verification table in order to authenticate the validity of users, which may risk the leakage of users' confidential information. After that, some researchers [1, 10, 16] proposed smart-card-based authentication schemes to achieve mutual authentication between a user and server. By applying the smart card mechanism, the server no longer needs to store the verification table in its database; on the contrary, most of verification parameters, such as users' personal information and secret parameters, are stored in the smart card. Users can use their own smart card to generate and send request messages to the server, which allows the server to recognize the validity of user. Traditionally, smart-card-based remote authentication has depended upon the verification of a user's identity and password; however, a user's identity can be easily ascertained by anyone. Furthermore, a user often tends to choose a short, simple, easy-to-remember, and auto-correlated string as his/her password (e.g., telephone number, birthday, or commemoration day); thus, the user's password may be guessed by someone such as a close friend or colleague. In order to reduce the risk of the user identity or password being compromised, several researchers have begun to employ biometric information as part of the verification of user validity, because biometric information (e.g. iris, fingerprint) is unique to each user and hard for others to guess or obtain. Since 2002, more and more studies have combined individuals' biometric information to achieve user authentication using smart cards [4, 12, 13]. However, all of the schemes in [4, 12, 13] are designed for the single-server architecture.

Taking into account the diversification of services, users may want to access different services from different service providers, which may cause users to register an account for each service provider in the single-server environment. Thus, He et al. [7] proposed a smart-card-based authentication scheme for the multi-server environment which allow users to register the system only one time for access to all service providers in the system. Besides, in order to reduce the risk of the user identity or password being compromised, some biometrics-based authentication schemes for the multi-server environment are proposed [3, 6, 17, 18]; however, Yang and Yang's [17] biometric password-based multi-server authentication scheme

has heavy computation cost because it applies so many modular exponentiation operations. In addition, in regard to Yoon and Yoo's scheme [18], He [6] found that it is vulnerable to privileged-insider attack, masquerade attack, and stolen smart card attack. Recently, Chuang and Chen [3] pointed out a common weakness of most biometrics-based schemes is inefficiency in addition to some security problems; thus, they proposed an improved scheme with higher efficiency and security under the assumption that all registered servers are trusted. However, both Choi [2] and Mishra [14] found that Chuang and Chen's scheme cannot resist various attacks and fails to preserve the forward secrecy.

In our work, we find that Chuang and Chen's scheme [3] also suffers from the stolen smart card attack and forgery attack as well as a privacy preservation problem. Besides, their scheme has improper biometric error detection based on hash function, which may cause a serious false negative problem such that a valid user cannot successfully log in and access servers. Therefore, in this paper, we propose a more secure and efficient version that not only resists well-known attacks but satisfies the essentials for a well-designed multi-server authentication scheme. In particular, compared with Chuang and Chen's scheme [3] and other biometrics-based schemes [4, 12, 13, 17, 18], our scheme overcomes the false negative problem in biometrics detection by adopting the functions defined in Dodis et al.'s literature [5], which feature fault tolerance in biometrics information.

The remainder of this paper is organized as follows. In Section 2, we briefly review Chuang and Chen's biometrics-based multi-server authentication scheme using smart cards and discuss its weaknesses. Section 3 introduces some requirements that our proposed scheme needs to achieve and reviews Dodis et al.'s secure sketch definitions used in our proposed scheme. Subsequently, our advanced biometrics-based multi-server authentication scheme is provided in Section 4, and security analyses of our proposed scheme are discussed in Section 5, followed by comparisons of relevant schemes in Section 6. Finally, our conclusions are shown in Section 7.

2 Review and Cryptanalyses of Chuang and Chen's Scheme

In 2014, Chuang and Chen [3] pointed out that the common weakness of most biometrics-based schemes is inefficiency in addition to some security problems. Hence, they proposed a hash-based scheme with higher efficiency as shown in Figure 1. Furthermore, the security of their scheme is based on the assumption that all registered servers are trusted. Though Chuang and Chen claimed that their scheme can enhance several security properties, we find that their scheme still has some security flaws. Thus, in this section, we analyze and describe its vulnerabilities as follows. Note that Chuang and Chen claimed that, in their scheme, an authorized server could be trusted under the assumption of trust computing; hence, herein, we also do not consider the possibility of a server being dishonest.

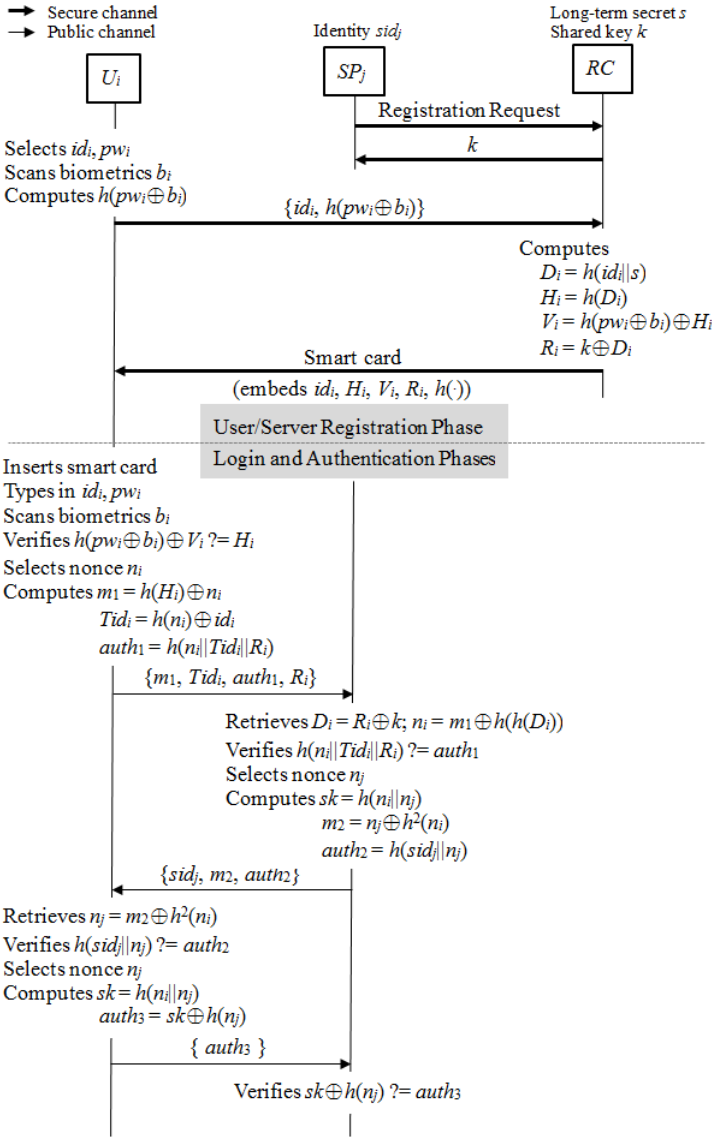


Figure 1: Review of Chuang and Chen's scheme

2.1 Stolen Smart Card Attack

In [3], Chuang and Chen claimed that if someone knows a valid user's parameters stored in the user's smart card, he/she cannot forge a valid message to pass authentication; however, if an attacker, U_A, steals the card from a U_i in some way and retrieves the information stored in it, he/she can easily forge a valid message and impersonate the user to log in to the system by the following procedures.

Step 1. U_A extracts the parameters $\{id_i, H_i, V_i, R_i, h(\cdot)\}$ from the smart card.

Step 2. U_A chooses a random nonce n_A and computes $m_{1A} = h(H_i) \oplus n_A$, $Tid_i = h(n_A) \oplus id_i$, and $auth_{1A} =$

$h(n_A||Tid_i||R_i)$, then U_A sends the fake login request message $\{m_{1A}, Tid_i, auth_{1A}, R_i\}$ to SP_j.

Step 3. After receiving the message from U_A, SP_j retrieves $D_i = R_i \oplus k$ and $n_A = m_{1A} \oplus h(h(D_i))$ by using its secret k shared with RC. Then SP_j authenticates the validity of U_A by checking $h(n_A||Tid_i||R_i)? = auth_{1A}$. It is obvious that the condition holds; thus SP_j is convinced that U_A is a valid user and accepts the login request.

Step 4. SP_j randomly chooses a nonce n_j and computes $m_2 = n_j \oplus h^2(n_A)$ and $auth_2 = h(sid_j||n_j)$. Then, SP_j sends a reply message $\{sid_j, m_2, auth_2\}$ to U_A.

Step 5. Upon receiving the reply message from SP_j, U_A can easily retrieve $n_j = m_2 \oplus h^2(n_A)$ and verify whether $h(sid_j||n_j)$ is equal to the received $auth_2$. Obviously, the condition is satisfied; thus, U_A computes $auth_{3A} = h(n_A||n_j) \oplus h(n_j)$ and sends $auth_{3A}$ back to SP_j.

Step 6. Obviously, the reply message $auth_{3A}$ can pass the verification by SP_j after it has received this message. Until now, the attacker U_A has successfully logged in to SP_j and negotiated a session key $sk = h(n_A||n_j)$ with SP_j for future communication.

As aforementioned in regard to attack procedures, in [3], anyone who obtains a lost smart card can easily forge a valid message to pass the authentication, share a common session key, and access the services in the system. Moreover, as discussed in [14], an attacker can use a stolen smart card and intercepted messages to mount a server spoofing attack.

2.2 Forgery Attack

If an attacker, U_A, is a valid but malicious user, he/she can use his/her smart card and the parameters $\{id_A, H_A, V_A, R_A, h(\cdot)\}$ stored in it to mount a forgery attack without using his/her real identity id_A via the following procedures.

Step 1. U_A generates a fake identity id_f with the same length as the output of the hash function $h(\cdot)$. Then, U_A chooses a random nonce n_A and computes $m_{1A} = h(H_A) \oplus n_A$, $Tid_f = h(n_A) \oplus id_f$, and $auth_{1A} = h(n_A||Tid_A||R_A)$. Finally, U_A sends a fake login message $\{m_{1A}, Tid_f, auth_{1A}, R_A\}$ to SP_j.

Step 2. Upon receiving the message from U_A, SP_j retrieves $D_A = R_A \oplus k$ and $n_A = m_{1A} \oplus h(h(D_A))$ by using its secret k. Then SP_j computes and verifies whether the equation $h(n_A||Tid_f||R_A) = auth_{1A}$ holds or not.

Obviously, the verification would be successful. Because SP_j does not verify the validity of Tid_f, it cannot detect a forgery attack that uses a forged identity. Afterwards, by performing Steps 3 to 6 in Subsection 2.1, U_A and SP_j can complete mutual authentication and share a common session key for future communication. Furthermore, as mentioned in [14], an attacker also can use a lost smart card to forge a fake message to log in to the server.

2.3 Hash Function Problem in Terms of Biometrics

In Chuang and Chen's scheme, a user's smart card stores the parameter $V_i = h(pw_i \oplus b_i) \oplus H_i$, which includes the user's biometric information b_i scanned at the time when the user registered with RC. In the login phase of their scheme, the smart card verifies the validity of the card holder by checking $h(pw_i \oplus b_i^*) \oplus V_i \stackrel{?}{=} H_i$, where b_i^* is the biometric information scanned at this time. As advocated by Chuang and Chen [3], Figure 1, a valid card holder would pass the verification; however, in fact, a hash function is sensitive and free from collision, and the biometric information scanned by the same user each time may be slightly different. That is, the mapping from input to output of a hash operation is one-to-one, so a subtle change of input must impact the output of the hash operation. As a result, it would be unsuitable to use the hash function to detect the biometric information, as it may prevent a valid user from passing the authentication in Chuang and Chen's scheme.

2.4 Non-provision of User Privacy

In [3], Chuang and Chen indicated that the information stored in a smart card is extractable. Based on this assumption, anyone can obtain a user's real identity directly from the user's smart card because the identity is stored inside it.

On the other hand, if an attacker, U_A, wants to trace the locations or information related to a specific user, he/she may collect all transmitted messages from different sessions. In login and authentication phases of Chuang and Chen's scheme, as shown in Figure 1, we find that a user always transmits the same parameter R_i in each session. Hence, an attacker can monitor the transmitted R_i of each session to trace a specific user even if he/she does not know the user's actual identity from the message $\{m_1, Tid_i, auth_1, R_i\}$ publicly transmitted in Login Phase.

As a result, in Chuang and Chen's method, users are not guaranteed their privacy.

3 Preliminaries

Here, we introduce the essentials that must be achieved by a well-designed multi-server authentication scheme using smart cards, and the definition used in our proposed scheme using biometrics.

3.1 Requirements

In order to design a secure and efficient smart-card-based multi-server authentication scheme, the following six considerations must be satisfied.

1) No verification table: A registration center should not store any verification table in its database for the security consideration.
2) Single registration: This is the major property that distinguishes a multi-server system from a single-server system. For convenience, users only need to register with the registration center one time; then they can access the services from any service provider in a multi-server system.
3) Freely choose password: Users can freely choose and change their passwords without requiring the involvement of a registration center, in order to decrease the system load.
4) Mutual authentication and session key agreement: Users and service providers need to authenticate each other in order to prevent security problems and negotiate a common session key and thereby keep their communications secret.
5) Security: The designed authentication scheme should not only withstand various attacks but also avoid the synchronization problem. In addition, it also should preserve user privacy.
6) Efficiency: Since a smart card cannot support heavy computation in general, the computation load of the smart card must be made as low as possible.

3.2 Secure Sketch

As discussed in Subsection 2.3, a hash function is sensitive and free from collision, and the biometric information scanned by the same user may be slightly different each time, so a hash function is not capable of detecting the validity of biometric information. In order to overcome the problem in [3], in our proposed scheme, we adopt the functions defined by Dodis et al. [5] to deal with related operations of biometric information.

In 2004, Dodis et al. [5] defined that an (M, m, m', t)-secure sketch is a randomized map $SS : M \rightarrow \{0,1\}^*$, where m is min-entropy and m' is the lower bound of average min-entropy. One of its properties is as follows:

For any given vector $b' \in M$ satisfying $dis(b, b') \leq t$, there is a deterministic recovery function Rec such that $Rec(b', SS(b)) = b$, where dis is a distance function. Because a sketch does not reveal the information about b and it needs to give another value b' close to b, the design is secure.

Based on this definition, we can set SS as an $(M, m, m+k-n, t)$-secure sketch and $SS(B) = SS(B; X) = B \oplus E(X)$ for any given $[n, k, 2t+1]$ error-correcting code E, where B is uniform, X is random variable, n is the length of strings, k is the dimension of the code, and t is the number of tolerated errors. Also, there is a decoder D of the code E, which can correct up to t errors, such that $D(B' \oplus SS(B; X)) = X$ if $dis(B, B') \leq t$. As a result, the recovery function Rec can be set as $Rec(B', SS(B; X)) = SS(B; X) \oplus E(D(B' \oplus SS(B; X))) = B$.

4 Proposed Scheme

In this section, in order to solve the weaknesses of Chuang and Chen's scheme discussed in Section 2, and to achieve greater security, we present an advanced anonymous and biometrics-based multi-server authentication scheme using smart cards. The notations used in our proposed scheme are listed in Table 1.

Table 1: Notations

Item	Description
U_i	A user i
SP_j	A service provider j
RC	A trusted registration center
id_i	The identity of U_i
sid_j	The identity of SP_j
pw_i	The password of U_i
b_i	The biometric information of U_i
x	The secret key of RC
y	The secret number of RC
$E(\cdot)$	The encoding function based on Dodis et al.'s definition [5] (i.e., the error-correcting code in Subsection 3.2)
$D(\cdot)$	The decoding function based on [5] (i.e., the decoder in Subsection 3.2)
$h(\cdot)$	A secure one-way hash function

As in Chuang and Chen's scheme, there are three kinds of participants: users (U_i's), service providers (SP_j's), and a trusted registration center (RC). One of responsibilities of RC is to manage all service providers; the other is to assign a smart card for each legitimate user U_i who has registered with RC successfully. Once a user obtains the smart card from RC, he/she can use it and his/her personal information, such as identity, password, and biometrics, to log in to the system and access services provided by service providers SP_j's in this system. Accordingly, our proposed scheme consists of five phases: server registration, user registration, login, authentication, and password change. Additionally, in the system initialization, RC generates its secret key x and a secret number y. The first four phases are illustrated in Figure 2.

4.1 Server Registration Phase

If a server SP_j wants to become an authorized server, it needs to send a registration request to RC. Once RC accepts the application provided by this server, it uses its secret key x and the secret number y to compute $k_1 =$

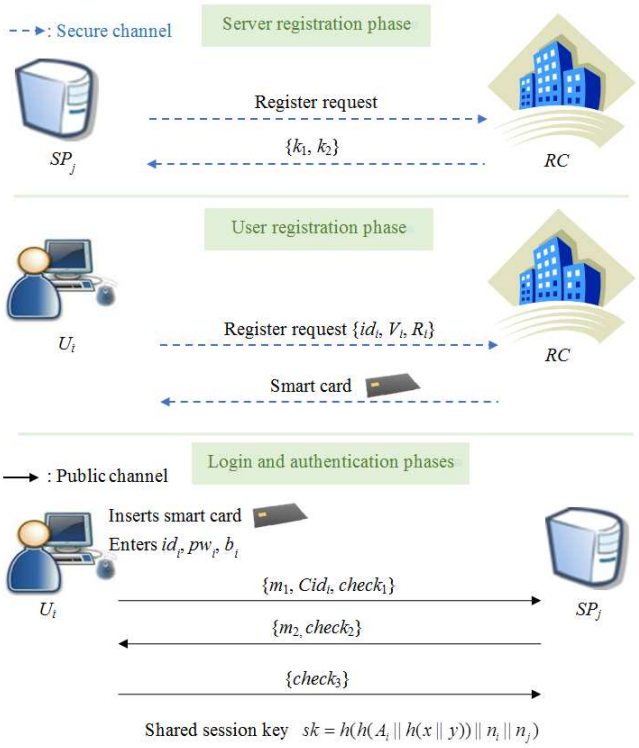


Figure 2: The flowchart of our proposed scheme

$h(sid_j||h(y))$ and $k_2 = h(x||y)$, then sends them back to the SP_j via a secure channel, where sid_j is the identity of SP_j. Afterwards, the server SP_j and RC share the secrets k_1 and k_2. Note that, in this system, each authorized server holds a unique secret k_1, which cannot be known by the others even though they have the same secret k_2. This is because that the secrets x and y are only known by RC.

4.2 User Registration Phase

In the course of the system's operation, a user U_i, who wants to access the resources of service providers in this system, must first register an account with RC. Then RC assigns a smart card embedded with some essential secret parameters to U_i. The detailed steps of user registration are described as follows.

Step R1. U_i generates his/her identity id_i and a password pw_i, and scans personal biometric information b_i (e.g., fingerprint) into the specific device. Then, U_i chooses a random number r_i to compute $\alpha_i = b_i \oplus E(r_i)$, $V_i = h(pw_i) \oplus \alpha_i$, and $R_i = h(pw_i \oplus r_i)$, and submits a registration request message $\{id_i, V_i, R_i\}$ to RC via a secure channel.

Step R2. After receiving the registration request from U_i, RC computes five parameters for U_i: $A_i = h(id_i||x)$, $B_i = h(id_i||R_i)$, $C_i = h^2(R_i) \oplus h(y)$, $D_i = h(R_i) \oplus A_i \oplus h(x||y)$, and $E_i = h(A_i||h(x||y)) \oplus h(R_i)$.

Step R3. RC stores the secret parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ into a smart card and sends it to the user U_i via a secure channel.

4.3 Login Phase

Once a user has been assigned a smart card from RC, he/she can use it to access any service at any time from the system by logging in to the corresponding server. Assume that a user U_i wants to log in to a server SP_j. First, he/she has to insert his/her smart card into a card reader, type in his/her id_i^* and pw_i^*, and scan personal biometrics information b_i^* into a specific device. Then, the smart card performs the following operations.

Step L1. The smart card computes $R_i^* = h(pw_i^* \oplus D(V_i \oplus h(pw_i^*) \oplus b_i^*))$ and verifies $h(id_i^*||R_i^*) \stackrel{?}{=} B_i$. If the verification is satisfied, as mentioned in Subsection 3.2, it indicates that the inputted b_i^* is close to the registered b_i in the user registration phase, and both inputted id_i and pw_i are correct. More precisely, the smart card is convinced that U_i is really the card holder and proceeds to the next step.

Step L2. The smart card randomly generates a nonce n_i and computes $h(y) = C_i \oplus h^2(R_i^*)$, $m_1 = h(sid_j||h(y)) \oplus n_i$, $Cid_i = D_i \oplus h(R_i^*) \oplus h(n_i)$, $G_i = E_i \oplus h(R_i^*)$, and $check_1 = h(h(sid_j||h(y))||n_i||G_i)$.

Step L3. The smart card sends $\{m_1, Cid_i, check_1\}$ as a login request message to SP_j.

4.4 Authentication Phase

Upon the login request message, SP_j and U_i execute the following steps to complete the mutual authentication and session key agreement.

Step A1. SP_j retrieves the nonce $n_i = m_1 \oplus k_1$ using its secret k_1 shared with RC, and checks the freshness of n_i. If the nonce n_i is fresh, SP_j subsequently uses the retrieved n_i, the received Cid_i from U_i, and its secret k_2 shared with RC to compute $A_i = Cid_i \oplus h(n_i) \oplus k_2$. Afterwards, SP_j computes and verifies whether $h(k_1||n_i||h(A_i||k_2))$ is equal to the received $check_1$. If the verification is failed, SP_j rejects the login request; otherwise, it confirms that U_i is valid and proceeds to the next step.

Step A2. SP_j randomly generates a nonce n_j and computes $m_2 = n_j \oplus n_i \oplus k_1$, $sk = h(h(A_i||k_2)||n_i||n_j)$, and $check_2 = h(sk)$. At last, SP_j sends a reply message $\{m_2, check_2\}$ to the user U_i.

Step A3. After receiving the reply message from SP_j, the smart card first retrieves the nonce n_j by computing $n_j = m_2 \oplus h(sid_j||h(y)) \oplus n_i$ and checks the freshness of n_j. If the nonce n_j is fresh, the smart card then uses it to compute the session key $sk = h(G_i||n_i||n_j)$ and verifies $h(sk) \stackrel{?}{=} check_2$. If the above authentication is satisfied, the smart card

ensures the validity of SP_j, which has received the correct n_i. Finally, the smart card computes and sends $check_3 = h(sk||n_j)$ to SP_j.

Step A4. Upon receiving $check_3$, SP_j computes and verifies whether $h(sk||n_j) = check_3$. If the equation holds, it indicates that U_i not only is a legal user but also has received the correct n_j generated by it; otherwise, the session is aborted.

Now, U_i and SP_j have shared a common session key sk such that they can use it to protect their future communication before the user logs out.

4.5 Password Change Phase

At any moment, if a user U_i wants to change his/her password, he/she needs to insert his/her smart card into a card reader, submit id_i and pw_i, and scan personal biometrics information b_i into a specific device for changing his/her old password pw_i to a new one pw_i^{new}. Then, the smart card executes the following steps.

Step P1. The smart card computes $\alpha_i = V_i \oplus h(pw_i)$, $r_i = D(b_i \oplus \alpha_i)$, and $R_i = h(pw_i \oplus r_i)$, and verifies whether $h(id_i||R_i)$ equals to the B_i stored in it. If they are equal, the smart card asks U_i to type in a new password; otherwise, the password change request is refused.

Step P2. After U_i types in his/her new password pw_i^{new}, the smart card uses it to compute

$$\begin{aligned} V_i^{new} &= V_i \oplus h(pw_i) \oplus h(pw_i^{new}), \\ R_i^{new} &= h(pw_i^{new} \oplus r_i), \\ B_i^{new} &= h(id_i||R_i^{new}), \\ C_i^{new} &= C_i \oplus h^2(R_i) \oplus h^2(R_i^{new}), \\ D_i^{new} &= D_i \oplus h(R_i) \oplus h(R_i^{new}), \\ E_i^{new} &= E_i \oplus h(R_i) \oplus h(R_i^{new}). \end{aligned}$$

Step P3. Lastly, the smart card replaces V_i, B_i, C_i, D_i, and E_i with V_i^{new}, B_i^{new}, C_i^{new}, D_i^{new}, and E_i^{new} stored inside it. Now, the user's password has been successfully changed without the help of RC.

5 Security Analyzes

In this section, we analyze the resistance to various attacks and the achievement of security requirements in our proposed scheme. Assume that an attacker, U_A, exists in the system who can not only control the whole public communication channel between users and service providers but also intercept, eavesdrop, or tamper with any transmitted message. We consider various different scenarios to provide detailed analyzes of our scheme below.

5.1 Resistance to Off-line Password Guessing Attack

Here, we illustrate two possible attackers' behaviors that would lead to an off-line password guessing attack as follows:

1) Stolen smart card of a user
If the attacker, U_A, has stolen a legal user's (U_i's) smart card, he/she may extract the parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it and try to mount an off-line password guessing attack; however, knowing the parameters in the card does not impact the security of U_i's password at all.

For V_i, it is calculated by the equation $V_i = h(pw_i) \oplus \alpha_i = h(pw_i) \oplus b_i \oplus E(r_i)$, where b_i is U_i's unique biometric information, and r_i is a random number secretly chosen by U_i. Actually, V_i belongs to a two-factor protection on r_i. In the regular login process of our scheme (i.e., Step L1), only inputting the correct pw_i and b_i can retrieve the random number r_i by computing $D(V_i \oplus h(pw_i) \oplus b_i)$. Note that there is no way to obtain or forge the user's unique biometrics information b_i. In other words, we also can treat V_i as a two-factor protection on pw_i. It is hard for U_A to detect whether he/she has guessed the correct password without the knowledge of b_i and r_i.

Similarly, for other stored parameters B_i, C_i, D_i, and E_i, their expressions all contain more than one element that U_A does not know in addition to the user's password pw_i, such as the user's identity id_i, long-term secrets x and y of RC, and the random number r_i. Hence, we can treat these parameters as multi-factor protections. It is hard for U_A to detect whether he/she has guessed the correct password without the knowledge of these elements. That is, id_i and r_i in $B_i = h(id_i||R_i)$; r_i and y in $C_i = h^2(R_i) \oplus h(y)$; and r_i, id_i, x, and y in $D_i = h(R_i) \oplus A_i \oplus h(x||y)$ and $E_i = h(A_i||h(x||y)) \oplus h(R_i)$. Consequently, our proposed scheme prevents the attacker from being about to guess a valid user's password in polynomial time from a user's stolen smart card.

2) Intercepting transmitted messages between users and service providers
If the attacker, U_A, intends to guess a specific user's password by intercepting messages transmitted between the user and an SP_j over the Internet, he/she will fail. This is because that, in our scheme, none of the transmitted messages are related to a user's password. As a result, it is impossible for an attacker to mount an off-line password guessing attack by collecting data transmitted over the Internet.

As discussed above, our proposed scheme can resist an off-line password guessing attack.

5.2 Resistance to Forgery Attack

Herein, to explain the resistance to forgery attack, we consider two cases: the general view and the privileged-insider view.

1) The general view
If the attacker, U_A, attempts to forge a valid message in order to log in to SP_j, he/she needs to forge the login messages as $m_1 = h(sid_j||h(y)) \oplus n_A$, $Cid_A = A_A \oplus h(n_A) \oplus k_2$, and $check_1 = h(h(sid_j||h(y))||n_A||h(A_A||k_2))$, where n_A and A_A are two random numbers chosen by him/her; however, it is difficult for an attacker to forge a valid login message without the long-term secrets k_1 and k_2 of the server. Even if the attacker knows the identity sid_j of the server, he/she still cannot know the secret k_1 without the knowledge of $h(y)$.

On the other hand, the attacker may intercept a legal user's (U_i's) login message $\{m_1, Cid_i, auth_1\}$, where $m_1 = h(sid_j||h(y)) \oplus n_i$, $Cid_i = A_i \oplus h(n_i) \oplus k_2$, and $check_1 = h(h(sid_j||h(y))||n_i||h(A_i||k_2))$, and try to forge another valid login message from it in order to impersonate that user. That is, U_A has to retrieve the user's information A_i from the intercepted message, to generate a nonce n_A, and to compute a fake login message $m_1^* = h(sid_j||h(y)) \oplus n_A$, $Cid_i^* = A_i \oplus h(n_A) \oplus k_2$, and $check_1^* = h(h(sid_j||h(y))||n_A||h(A_i||k_2))$ to SP_j. Obviously, it is computationally infeasible for U_A to retrieve the user's information A_i without the knowledge of the long-term secrets k_1 and k_2 of the server. Consequently, the forged message will be refused by SP_j in Step A1. Hence, U_A cannot forge a valid login message to impersonate the user U_i.

In addition, as mentioned in Subsection 5.1, most parameters stored in a smart card have the characteristic of multi-factor protection. Even if U_A has stolen a legal user's smart card and extracted the secret parameters $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it, he/she still cannot forge a valid message to log in to SP_j without knowing $h(y)$, pw_i, r_i, and U_i's biometric information b_i.

2) The privileged-insider view
If the attacker, U_A, is a valid but untrusted user, he/she may use his/her own parameters $\{V_A, B_A, C_A, D_A, E_A, h(\cdot)\}$ stored in his/her smart card to conduct the following forgery attack. U_A first computes $R_A = h(pw_A \oplus D(V_A \oplus h(pw_A) \oplus b_A))$ by using his/her pw_A and b_A, then U_A uses R_A and stored C_A to retrieve $h(y)$ by computing $h(y) = h^2(R_A) \oplus C_A$. Once U_A obtains the secret $h(y)$, he/she may try to mount a forgery attack, as mentioned in the third paragraph of Case 1, by taking a stolen smart card of a valid user U_i and retrieving the secrets $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it. Fortunately, as shown in the previous case (i.e., case 1), it is also impossible for U_A to forge a valid login message $\{m_1^*, Cid_i^*, auth_1^*\}$ to masquerade as another legal user without the knowledge of their pw_i and b_i, so the fake login message can still be detected by SP_j in Step A1.

The same holds, if U_A is a valid but dishonest service provider who has stolen a smart card of U_i and extracted the stored information. Though U_A additionally knows the secrets $k_1 = h(sid_j||h(y))$ and $k_2 = h(x||y)$, he/she still has no way to calculate $R_i = h(pw_i \oplus D(V_i \oplus h(pw_i) \oplus b_i))$ without the knowledge of pw_i and b_i. Hence, U_A still cannot impersonate a user by stealing a smart card to send a legal request in Step L3.

As mentioned above, our proposed scheme does not suffer from a forgery attack.

5.3 Resistance to Server Spoofing Attack

In regard to a server spoofing attack, if the attacker, U_A, tries to cheat a user, U_i, who sends a login request, he/she needs to forge and reply a valid message to U_i; however, U_A cannot impersonate a legal server, SP_j, to send a valid message since he/she does not have the secrets k_1 and k_2 of SP_j to extract the correct n_i and A_i from the intercepted login message $\{m_1, Cid_i, auth_1\}$ of U_i in Step A1. Hence, U_A cannot succeed in this attempt.

On the other hand, if the attacker is a valid service provider, SP_A, he/she still cannot masquerade as another service provider, SP_j, to fool a user, U_i, by forging an acknowledgement message in Step A2. The reason is that even though each valid server holds the same $k_2 = h(k||y)$, it is not the same as holding $k_1 = h(sidj||h(y))$, a shared key only known to SP_j and RC, is not equal. Thus, SP_A is unable to reply a valid message to U_i without knowing SP_j's k_1. As a result, there is no way for SP_A to impersonate another server to communicate with users.

In addition, we assume that the malicious SP_A has got a user's (U_b's) smart card and tries to know the k_1 of another SP_j by using the retrieved parameters $\{V_b, B_b, C_b, D_b, E_b, h(\cdot)\}$ from the stolen card. The main purpose is to obtain the partial secret $h(y)$ due to $k_1 = h(sid_j||h(y))$, where sid_j is the public identity of SP_j and y is a long-term secret of RC; however, it is computationally hard for SP_A to retrieve $h(y)$ from the stolen card, which can only be acquired by the card holder who has the correct password and personal biometric information. Thus, SP_A is unable to obtain other servers' keys k_1's without $h(y)$ to fool users.

No matter how the attacker masquerades as another server, his/her attempts will fail. As a result, our proposed scheme can resist a server spoofing attack.

5.4 Resistance to Stolen Smart Card Attack

As mentioned in Subsections 6.2 and 6.3, no matter whether the attacker is an outsider or not, he/she cannot use the stolen smart card to impersonate any other person or server. Furthermore, the attacker cannot know the actual owner of the stolen card in addition to the password guessing attack in Subsection 5.1. Hence, stealing a user's smart card does not enable an attacker to acquire the user's private information and perform illegal behavior. That is to say, our proposed scheme can fully prevent a stolen smart card attack.

5.5 Resistance to Stolen-verifier Attack

With regard to this attack, the attacker may try to steal the verification table stored in the database of *RC* in order to engage in illegal behavior, such as a user impersonation attack in order to access the services of service providers; however, our scheme needs not to be worry about the stolen-verifier attack by adopting smart-card-based authentication. In our scheme, *RC* does not store any verification table regarding users' accounts; thus, the attacker is unable to gain the information that would be used to impersonate another user successfully. In other words, the attacker cannot succeed in such an attempt by using the process discussed in Subsection 5.2. As a result, a stolen-verifier attack is infeasible in our proposed scheme.

5.6 Resistance to Privileged-insider Attack

It is possible that there an inside attacker, U_A, may have the right to obtain the message sent from users in the user registration phase over the secure channel or information from the database of the registration center. Though the system claims that it can be trusted, a privileged-insider may be able to use his/her privileges to obtain a user's identity, password, and biometric information; however, in our scheme, the user U_i sends $\{id_i, V_i, R_i\}$ as a registration request as shown in Step R1 of the registration phase, where $V_i = h(pw_i) \oplus b_i \oplus E(r_i)$, $R_i = h(pw_i \oplus r_i)$, and r_i is a random number. The password and biometric information are protected by a secure one-way hash function, which means that it is computationally infeasible for an insider to know a user's password and biometric information directly. Furthermore, as discussed in Subsection 5.1, a user's password cannot be guessed by others except for the user himself/herself. In addition, each user's biometric information is unique and held by himself/herself; thus, the insider has difficulty obtaining users' biometric information. Therefore, even if there is a privileged member in the system, he/she is unable to obtain a user's private information and engage in wrongdoing.

5.7 Resistance to Replay Attack

Our scheme is free from the replay attack, because we adopt random nonces instead of timestamps that could result in a time synchronization problem. During the protocol run, the freshness of transmitted messages $\{m_1, Cid_i, check_1\}$ from a user and $\{m_2, check_2\}$ from a service provider must be verified by checking n_i and n_j in Steps A1 and A3, respectively. If the attacker replays the message intercepted in the previous session, our scheme will quickly detect that the involved random nonce is invalid, because the random nonce must be different in each session. Obviously, the replayed message would not pass authentication in our scheme. As a result, our proposed scheme can withstand the replay attack.

5.8 No Hash Function Problem in Terms of Biometrics

As discussed in Subsection 2.3, Chuang and Chen adopted the hash function to detect a user's biometric information, but this would lead to a serious false negative problem as a valid user may not pass verification by using his/her own smart card. In order to overcome this problem, in our scheme, we do not use the hash function but rather encoding and decoding functions defined by Dodis et al. [5] to detect a user's biometric information. These encoding and decoding functions have fault tolerance in biometric information, as mentioned in Subsection 3.2, which allows a user to pass authentication even if the biometric information that he/she scanned each time is slightly different from the original one scanned during the registration phase.

Hence, our improved scheme can avoid such a problem in detecting users' biometric information.

5.9 Preservation of Known-key Security

In our proposed scheme, the session key is computed as $sk = h(h(A_i||k_2)||n_i||n_j)$, which involves $A_i = h(id_i||x)$, $k_2 = h(x||y)$, and the random nonces n_i and n_j chosen by U_i and SP_j in each session, respectively. The nonces are delivered in the form $m_1 = h(sid_j||h(y)) \oplus n_i = k_1 \oplus n_i$ and $m_2 = n_j \oplus n_i \oplus k_1 = n_j \oplus n_i \oplus h(sid_j||h(y))$, which implies that only the valid SP_j and U_i have the secret k_1 and $h(y)$ can retrieve the correct nonces n_i and n_j. Furthermore, we assume that the long-term secrets x and y of *RC* are only known by *RC* itself and are not stored in the verification table or in users' smart cards based on Shannon's and Kerckhoffs' Theorems [8, 9, 15]. Consequently, the attacker cannot obtain the long-term secrets of *RC* from its database or a user's smart card. In other words, if the kth session key is compromised accidentally by an attacker, it will not reveal the confidential content of messages nor the session keys negotiated in previous and following sessions, because the session key is changed by the nonces n_i and n_j.

Table 2: Security comparisons of our proposed scheme with relevant schemes

Property	Ours	[3]	[18]	[17]	[4]	[13]	[12]
Off-line password guessing attack	Yes	Yes	Yes	Yes	No	Yes	Yes
Forgery attack	Yes	No	No	Yes	No	Yes	Yes
Server spoofing attack	Yes	No	Yes	No	No	Yes	Yes
Stolen smart card attack	Yes	No	No	Yes	No	Yes	Yes
Stolen-verifier attack	Yes	Yes	Yes	Yes	Yes	No	Yes
Privileged-insider attack	Yes	Yes	No	No	No	No	No
Replay attack	Yes	Yes	Yes	Yes	Yes	No	Yes
No hash function problem in terms of biometrics	Yes	No	Yes	Yes	Yes	No	No

Yes: The scheme actually satisfies the property or resists the attack;
No: The scheme does not satisfy the property.

As a result, our proposed scheme can achieve known-key security in the session key establishment.

5.10 User Privacy Preservation

Here, we discuss various aspects of user privacy preservation:

1) Privacy of user's identity and location
If the attacker, U_A, wants to know a user's (U_i's) identity, he/she may capture the transmitted message $\{m_1, Cid_i, check_1\}$ sent by U_i, and try to retrieve U_i's identity from $Cid_i = D_i \oplus h(R_i) \oplus h(n_i) = h(id_i||x) \oplus h(x||y) \oplus h(n_i)$ or $check_1 = h(h(sid_j||h(y))||n_i||G_i) = h(h(sid_j||h(y))||n_i||h(h(id_i||x)||h(x||y)))$. In our scheme, however, U_i communicates with SP_j anonymously such that even SP_j cannot know the user's actual identity. Besides, the identity involved in Cid_i and $check_1$ is protected by a secure one-way hash function as well as the secrets x and y of RC. It is computationally infeasible for the attacker to retrieve U_i's real identity, id_i, from Cid_i and $check_1$. Similarly, even if U_A has obtained a user's smart card and extracted the stored secrets $\{V_i, B_i, C_i, D_i, E_i, h(\cdot)\}$ from it, he/she still cannot know the user's actual identity, id_i.

   In addition, even if U_A focuses on tracking U_i's location without knowing U_i's identity, he/she is unable to do so. This is because the message transmitted from a user is generated dynamically in each session by adopting a random nonce. Hence, it is difficult for an attacker to track a specific user's real location.

   As a result, our scheme can preserve the anonymity and untraceability of users.

2) Privacy of user's biometric information
If U_A wants to know a user's biometric information, he/she needs to obtain the user's smart card and try to retrieve the user's biometric information from V_i stored in the smart card; however, the user's biometric information, b_i, is encrypted by the encoding function as shown in Step R1; thus, U_A cannot obtain b_i directly. Besides, it is hard for U_A to guess the correct b_i in polynomial time from the stored parameters of the stolen smart card, since each user holds unique biometric information. Consequently, the privacy of a user's biometric information is guaranteed.

6 Comparisons

In this section, we provide security, functionality, and performance comparisons of our scheme with other biometrics-based authentication schemes [3, 4, 12, 13, 17, 18].

First, we summarize security comparisons between our scheme and other related schemes in Table 2. As discussed in Section 5, our scheme can prevent the listed attacks and avoid the hash function problem in terms of biometrics. In regard to an off-line password guessing attack, all of the related schemes can resist it except for [4]. In [4], if an attacker has obtained a user's smart card and extracted the stored secrets in some way, then he/she can easily guess the user's password in polynomial time. In [3, 4, 18], an attacker can forge a valid message to cheat servers by using the stolen smart card of a valid user as well as intercepted messages, so these schemes are unable to resist forgery and stolen smart card attacks. Similarly, in regard to a server spoofing attack, in [3, 4], an attacker can use a stolen smart card and intercepted messages to impersonate a valid server and send a legal message to a user. In [17], however, since each server holds the same secret shared with the RC, a valid but dishonest server can impersonate another server to communicate with users. In Li et al.'s scheme [13], because this scheme stores the verification table in database, it cannot prevent a stolen-verifier attack. Besides, in their scheme, users and the server never check the freshness of transmitted messages during their verification procedures, so their scheme cannot resist a replay attack.

Table 3: Functionality comparisons of our proposed scheme with relevant biometrics-based authentication schemes

Functionality	Ours	[3]	[18]	[17]	[4]	[13]	[12]
No verification table	Yes	Yes	Yes	Yes	No	No	Yes
Single registration	Yes	Yes	Yes	Yes	N/A	N/A	N/A
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Preservation of known-key security	Yes	Yes	Yes	Yes	N/A	Yes	N/A
Privacy of user's identity and location	Yes	No	No	No	No	No	No
Privacy of user's biometric information	Yes	Yes	No	Yes	No	No	Yes
No time synchronization	Yes	Yes	Yes	No	Yes	Yes	Yes
Quick error detection	Yes	Yes	No	Yes	Yes	No	No
Freely choose and change password	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes	No	Yes	No

Yes: The scheme actually satisfies the functionality;
No: The scheme does not satisfy the functionality;
N/A: The scheme does not consider or be applicable to the functionality.

It is noteworthy that the schemes in [4, 12, 13, 17, 18] cannot prevent a privileged-insider attack. In these schemes, a privileged-insider can know a user's identity, password, and biometric information directly, because the user sends his/her personal information as the registration request in plaintext form to the *RC*. Once the privileged-insider obtains a user's identity, password, and biometric information, he/she can impersonate the user to do anything. Furthermore, the schemes in [3, 12, 13] have a hash function problem in terms of biometrics, because these schemes use the hash function to detect whether a user's biometric information is correct or not, which will lead to a serious false negative problem where a valid user may be denied verification as discussed in Subsection 2.3.

Second, Table 3 summarizes functional comparisons between our scheme and other relevant schemes. All of schemes provide mutual authentication. As analyzed in Section 5, our proposed scheme achieves all functionality requirements. In [4, 13], because each user's biometric template is stored in the system, both schemes cannot achieve the no verification table requirement. Consequently, these schemes cannot preserve the privacy of users' biometric information. In [4, 12, 13], schemes were designed for single-server architecture, so the property of single registration is inapplicable to them.

Particularly, in regard to the privacy preservation of a user's identity and location, all of the related schemes [3, 4, 12, 13, 17, 18] are unable to preserve this property because, in these schemes, a user's identity can be obtained from a stolen smart card or transmitted messages. In Yoon and Yoo's scheme [18], a user's biometric template is stored in the user's smart card directly. Once an attacker gets the smart card, he/she can easily obtain the user's biometric information. Hence, the scheme in [18] cannot preserve the privacy of users' biometric information.

Furthermore, all of the schemes adopt the nonce mechanism rather than timestamps to resist the replay attack except for Yang and Yang's scheme [17]. Thus, only Yang and Yang's scheme suffers from the time synchronization problem. In regard to quick error detection in [12, 13, 18], a smart card can verify biometrics quickly but cannot detect the password in time, because it has to wait for the server to authenticate the messages in order to know the correctness of the password. Thus, these schemes do not really achieve this property.

Lastly, performance comparisons of the login and authentication phases of our proposed scheme and other relevant schemes are shown in Table 4. Considering that the computation cost of smart cards is limited, attention should be paid to the performance analysis of the proposed scheme. First, let us define the notations used in Table 4. T_h is the computation time for performing a one-way hash function once; T_D refers to the computation time of one decoding operation based on Dodis et al.'s definition [5]; T_{ecc} refers to the computation time of one elliptic curve operation; T_e indicates the computation time of one modular exponentiation operation; and T_f indicates the computation time for executing fuzzy extractor once. In addition, we ignore the cost of the exclusive-OR operation, because its time complexity is much lower than the above operations. On the other hand, as cost implementation in [3], the order of time complexity is $T_e >> T_{ecc} >> T_h$. We assume that the costs of T_D and T_f are low.

Table 4 shows that, in our proposed scheme, the total computation cost of login and authentication phases is $17T_h+1T_D$. For the multi-server environment, our scheme obviously is more efficient than Yoon and Yoo's [18] and Yang and Yang's [17] schemes; however, our scheme costs a little more than Chuang and Chen's [3]. It is still reasonable, though, because our scheme can use the encoding and decoding function defined in [5] to greatly reduce the false negative problem related to biometrics error de-

Table 4: Performance comparisons of the login and authentication phases of our proposed scheme and other relevant schemes

Participant	Ours	[3]	[18]	[17]	[4]	[13]	[12]
Registration center	X	X	$7T_h$	X	X	X	X
Service provider	$6T_h$	$8T_h$	$2T_{ecc}+$ $5T_h$	$3T_e+$ $3T_h$	$5T_h$	$6T_h$	$3T_h$
User	$11T_h+$ $1T_D$	$9T_h$	$2T_{ecc}+$ $5T_h$	$2T_e+$ $5T_h+$ T_f	$5T_h$	$7T_h$	$4T_h$
Total	$17T_h+$ $1T_D$	$17T_h$	$4T_{ecc}+$ $17T_h$	$5T_e+$ $8T_h+$ T_f	$10T_h$	$13T_h$	$7T_h$

X: There is no computation cost for this entity in the login and authentication phases.

tection. Moreover, although our scheme has higher cost than the schemes in [4, 12, 13], our scheme supports the multi-server environment, resists most potential attacks, and has more functionalities, as shown in Tables 2 and 3. Therefore, it is worth increasing the cost in order to provide more functionalities and higher security.

As a result, our scheme not only ensures security but also maintains functionality and efficiency better than other biometrics-based schemes.

7 Conclusions

In this work, we find that Chuang and Chen's scheme cannot resist stolen smart card and forgery attacks and cannot guarantee user privacy. In particular, their scheme has an improper design in regard to biometrics error detection. Thus, we propose an improved biometrics-based multi-server authentication scheme using smart cards. As shown in our security analyses and comparisons, the proposed scheme not only remedies the flaws of Chuang and Chen's scheme but also prevents vulnerability to various attacks and achieves the necessary requirements. Furthermore, our proposed scheme has lower computational cost used for authentication, as shown in Table 4. Consequently, our proposed scheme is not only suitable for applying biometrics detection but also is efficient and robust against most security attacks.

References

[1] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient and practical solution to remote authentication: smart card," *Computers and Security*, vol. 21, no. 4, pp. 372–375, 2002.

[2] Y. Choi, "Security enhanced anonymous multi-server authenticated key agreement scheme using smart card and biometrics," *IACR Cryptology ePrint Archive*, pp. 1–11, 2014.

[3] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[4] A. K. Das, "Analysis and improvement on an efcient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[5] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, LNCS 3027, pp. 523–540, Springer, 2004.

[6] D. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology ePrint Archive*, pp. 1–9, 2011.

[7] D. He, W. Zhao, and S. Wu, "Security analysis of a dynamic ID-based authentication scheme for multi-server environment using smart cards," *International Journal of Network Security*, vol. 15, no. 5, pp. 350–356, 2013.

[8] A. Kerckhoffs, "La cryptographie militaire," *Journal des Sciences Militaires*, vol. January, no. IX, pp. 5–38, 1883.

[9] A. Kerckhoffs, "La cryptographie militaire," *Journal des Sciences Militaires*, vol. February, no. IX, pp. 161–191, 1883.

[10] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 204–207, 2004.

[11] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[12] C. T. Li and M. S. Hwang, "An efcient biometrics-based remote user authentication scheme using smart

cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[13] X. Li, J. W. Niu, J. Ma, W. D. Wang, and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.

[14] D. Mishra, "Cryptanalysis of multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," arXiv e-print service, Cornell University, pp. 1–8, 2014.

[15] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[16] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," *Computers and Security*, vol. 18, no. 8, pp. 727–733, 1999.

[17] D. Yang and B. Yang, "A biometric password-based multi-server authentication scheme with smart card," in *Proceedings of IEEE 2010 International Conference on Computer Design and Applications (ICCDA'10)*, pp. 25–27, Qinhuangdao, China, 2010.

[18] E. J. Yoon and K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2010.

Chin-Chen Chang received his Ph.D. degree in computer engineering 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 title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. He is a Fellow of IEEE and a Fellow of IEE, UK. His research interests include database design, computer cryptography, image compression and data structures.

Wei-Yuan Hsueh received the B.S. degree in computer science and information engineering from National United University, Miaoli, Taiwan in 2012. She received the M.S. degree in computer science and information engineering from National Chung Cheng University, Chia-Yi, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and wireless communications.

Ting-Fang Cheng received her Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan in 2013. She received the B.S. and M.S. degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan in 2005 and 2007, respectively. Now she is as a Postdoctoral Fellow in information engineering and computer science, Feng Chia University, Taichung, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, cloud computing, and information hiding.

A Survey of Data Distortion Watermarking Relational Databases

Ming-Ru Xie[2], Chia-Chun Wu[3], Jau-Ji Shen[2], Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan (R.O.C.)
(Email: mshwang@asia.edu.tw)
Department of Management Information System, National Chung Hsing University[2]
Department of Industrial Engineering and Management, National Quemoy University[3]
No. 1, University Rd., Jinning Township, Kinmen County 892, Taiwan (R.O.C.)
Department of Medical Research, China Medical University Hospital, China Medical University[4]
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan (R.O.C.)
(Received Oct. 18, 2015; revised and accepted Jan. 19, 2016)

Abstract

Watermarking relational database is a technique which can provide ownership protection and temper proofing for relational databases. Although it has been developed over ten years, it is still not popular. For attracting more people to study this technique, we introduce it in detail in this paper. The main contributions of this paper include: 1) To the best of our knowledge, this is the first paper which specially surveys data distortion watermarking relational databases; 2) We define a new requirement analysis table for data distortion watermarking relational databases and use it to analyze important and the newest research of data distortion watermarking relational databases; 3) We explain background knowledge of watermarking relational databases, such as types of attacks, requirements, and basic techniques.

Keywords: Database security, database watermarking, ownership protection, watermarking relational databases

1 Introduction

Since the first set of relational database product appeared in 1981, it has gradually become an important software system which is used to store data for a private company and government institutions. A private company uses the relational database to store customer data, ordering data, shipment data of a company, etc. The government uses it to store project data, tax data, etc.

In early stage, the relational database can only store data, and then Data Warehouse and Data Mining technology appear, which make the relational database can analyze and find out the hidden special relationships among data in the database by data mining; and these are available for a company to make decision [13]. It can be seen in the future, "data" will be an important asset. How do we protect the data stored in the relational database? Is it safe enough for relational databases nowadays? We'll discuss this issue in the following paragraphs.

The data stored in the relational database is the same as images, videos, etc., and they are all digital data; they all have a characteristic that can be duplicated, and moreover, the appearance of Internet makes digital data can be easily transferred to others through the Internet, so that these issues that result in theft problems are getting worse and worse. Although the relational database has an authority control security mechanism which can limit an illegal user to access database, in recent years, the news about a legal user stealing and selling data still sometimes happens. When a legal user steals the data in the relational database and sells it, the theft party claims that the data belongs to him; how can we prove whom do the data belongs to?

In addition, due to the development of data mining technology, data owners can provide the relational database to a data mining company for data mining [12]. In the process of transferring the relational database to the recipient, the data may be stolen and tampered by an attacker, and then the attacker transfers the tampered relational database to the recipient. In this case, how do we prove that the data in the relational database is not tampered? Based on the above, we can embed watermark information into the relational database in order to prove ownership and tamper proofing for relational databases. This kind of technique is known as watermarking relational databases [1, 10, 13].

The rest of our paper is arranged as follows: Section 2

briefly introduces the history of watermarking relational databases. We explain background knowledge of watermarking relational databases in Section 3. Section 4 surveys in detail important and the newest research of data distortion watermarking relational databases and elaborates requirement analysis tables for them. In Section 5, we compare techniques of Section 4 and conclude by some issues for watermarking relational databases, and then propose future work.

2 Related Work

In digital media, such as videos, images, the technique which embeds a digital watermark to prove copyright has been developed for many years. In 2000, Khanna et al. proposed a concept to use a digital watermark in a database in order to protect a database of map information [14], and then many scholars began to research in this area. Finally in 2002, Agrawal and Kiernan proposed the first implementation method [1]. They calculated one LSB of one numeric attribute of some tuples in the relational database, and this is where they intend to embed the watermark. Next, they embedded the watermark into the selected LSB.

The research for watermarking relational databases can be grouped into two kinds: Data distortion watermarking relational databases and data distortion-free watermarking relational databases [3, 6, 22]. The research proposed by Li et al. [18], Yang et al. [28], Mehta et al. [20], Ali et al. [2], Hanyurwimfura et al. [9], Prasannakumari et al. [24], and Melkundi et al. [21], all belong to data distortion watermarking relational databases.

Latest important research in data distortion watermarking relational databases is that Kamran et al. [12] proposed a robust, distortion minimizing technique. Their technique includes three main steps: The first step includes Data Partitioning, Selection of Data Set for Watermarking and Hash Value Computation. Its purpose is to pick the position used to embed the digital watermark. Data Partitioning uses Algorithm 1 (Get_Partitions) to partition the data. Selection of Data Set for Watermarking uses Algorithm 2 (Get_Data_Selection_Threshold) to establish threshold for singling out the data sets from data partitions in the first step, and then it uses Algorithm 3 (Get_Even_Hash_Value_Data Set) to decrease these data sets. After first step, we will get data sets which can be used to embed the watermark. The second step is Watermark Embedding, and it uses Algorithm 4 (Embed_Watermark) to embed the watermark. The third step is Watermark Decoding, and it uses Algorithm 5 (Detect_Watermark) to detect the embedded the watermark. This algorithm begins to detect the watermark after it uses Algorithms 1, 2 and 3 to find out data sets are embedded with the watermark.

A sub-domain called reversible watermarking relational databases was proposed in 2006. It comes from the image and belongs to data distortion watermarking relational databases. Generally speaking, after embedding digital watermarks, the data will distort, but this technique can recover the raw data. Zhang et al. proposed the first scheme in 2006 [29]. By expansion on a data error histogram, they accomplished reversible watermarking relational databases. However, it is not robust enough to resist violent attacks [10, 29]. Latest important research was proposed by Iftikhar and Kamran et al. [10]. They proposed an RRW technique. RRW includes four steps: The first step is Watermark preprocessing. It selects the features ready to embed the digital watermark, and then generates the watermark via Genetic Algorithm. The second step is Watermark encoding, and it uses Algorithm 1 (Watermark Encoding) to embed the watermark into selected features. The third step is Watermark decoding, and it uses Algorithm 2 (Watermark Decoding) to retrieve the watermark. The fourth step is Data recovery, and it uses Algorithm 3 (Data Recovery) to recover raw data.

Next, we discuss data distortion-free watermarking relational databases. The first scheme in this domain should be Li et al.'s scheme. Via parameters, the primary key and the secret key, they calculate the hash value of tuples and primary key, respectively. And then they determine the locations used to embed the digital watermark via the hash values. Their digital watermark is produced via the hash values and the secret key [17]. In 2006, Tsai proposed that digital watermark can be generated via using images and features of the relational database [8, 27]. Recent research in this domain is proposed by Camara et al.. Their technique first partitions the data into many square matrix groups, and then computes these groups in order to generate the watermark, and then encrypts the watermark in order to get the watermark certification. Eventually, a CA (Certification Authority) will enroll the watermark certification. At CA, we can get the original watermark from the watermark certification. After we retrieve a new watermark from the database, we can compare it with the original watermark in order to check the integrity of data in the relational database [5].

3 Background

3.1 Types of Watermarks

A digital watermark is a kind of digital signature of digital media, and it can represent the author. It is grouped into two kinds:

1) Invisible Watermark: It embeds digital watermarks which can represent the author into digital media, and tries not to affect the quality of digital media. Because the human senses cannot become aware of very tiny changes, the naked eye cannot distinguish whether the embedded digital media has digital watermarks or not.

2) Visible Watermark: Typically, it uses a logo or text as a watermark, and then these watermarks can be

identified with the naked eye [19]. Its advantage is without going through any operation, and the watermark is very clear and visible; its disadvantage is it would destroy the quality of the original digital media.

3.2 Types of Attacks

After embedding the watermark into the relational database, it might suffer from assorted purposeful and unwilled attacks. We explain these possible attacks in the following paragraphs [4, 8, 12]:

1) Insertion attack: The attacker inserts new tuples into the relational database in order to eliminate the digital watermark.

2) Alteration attack: The attacker eliminates the digital watermark by modifying the value of tuples in the relational database. As long as the attacks have changed the value of tuples, these all belong to this category, for example, Bit flipping attack.

3) Deletion attack: The attacker eliminates digital watermarks by deleting tuples in the relational database.

Above mentioned attacks are basic attacks. Advanced attacks are as follows:

1) Multifaceted attack: A sophisticated attacker would mix assorted attacks, such as insertion attack, deletion attack and alteration attack to eliminate the digital watermark in the relational database.

2) Additive attack: The attacker fakes his own ownership of the relational database by embedding his digital watermark into the relational database.

3) Subset attack: The attacker only modifies or deletes a subset of tuples or attributes in the relational database in order to eliminate the digital watermark.

4) Superset attack: The attacker adds new tuples or attributes into the watermarked relational database in order to influence retrieval of the digital watermark.

5) Subset reverse order attack: The attacker changes the locations or order of tuples or attributes in the watermarked relational database in order to eliminate the digital watermark.

6) Mix-and-Match attack: The attacker collects related information from a different relation to build his own relation.

7) Brute force attack: The attacker uses programs to guess at the possible private parameters, for example, a secret key. This attack will try all possible private parameters until it finds the correct answer. If the length of private parameters is long enough, then this attack can be prevented.

8) Benign update: A relational database embedded with the digital watermark may affect the embedded digital watermark under usual insertion, deletion and modification, so that the watermark cannot be retrieved.

9) Invertibility attack: The attacker finds the fake watermark in the watermarked database, but this fake watermark is created by a random sequence.

3.3 Requirements

According to many literatures we referred to, a technique for watermarking relational databases has the following requirements [8, 13, 15, 19]:

1) Robustness: A digital watermark must be able to resist malicious attacks. After the attack, it will not be destroyed easily, and the embedded digital watermark still be extracted.

2) Unambiguity: The digital watermark retrieved by this technique must clearly identify its owner.

3) Security: Selection of the position used to embed the digital watermark is determined by some secret parameters, for example, a secret key. These secret parameters must keep secret, and they only can be known by certain people, e.g. database owner.

4) Blindness: The digital watermark must be retrieved without the original relational database or digital watermark information.

5) Imperceptibility: The embedded digital watermark must be indistinguishable.

6) Usability: After embedding the digital watermark, the data in the relational database is still usable; the best situation is this technique does not lead to the distortion of raw data.

We think that a technique for watermarking relational databases needs to meet above mentioned six requirements, and then it will be an effective watermarking relational databases. After we survey above mentioned research, we find they only define requirements, but they don't analyze techniques for watermarking relational databases by these requirements. Therefore, it is hard to compare them. Next, we try to define a new requirement analysis table for data distortion watermarking relational databases. As far as we know, this is the first requirement analysis table which uses these requirements to analyze techniques for watermarking relational databases, so it can bring a lot of help for comparison. The explanation and the format of the requirement analysis table is as in Table 1.

About robustness, we list all attacks used in their experiments. About unambiguity, security, and blindness, we use Yes or No to show if this technique meets this requirement or not. About imperceptibility, we believe that

Table 1: The explanation and the format of the requirement analysis

Proposed Scheme	The name of proposed scheme.
Robustness	The attacks used in their experiments.
Unambiguity	Yes or No.
Security	"Yes" means it has secret parameters.
Blindness	Yes or No.
Imperceptibility (%)	The discontinuous degree of the watermark bits in the database.
Usability	The amount of data distortion.

the watermark bits are more scattered in the database, that is, the discontinuous degree of the watermark bits in the database is higher, and then this technology is better. About usability, we believe that the lower the amount of data distortion, the better this technology. The following techniques we survey will focus on the six points to explain.

3.4 Watermarking Relational Databases

watermarking relational databases is a technique which embeds an invisible digital watermark into the relational database. It includes two primary steps [16], watermark embedding stage and watermark retrieve stage. In Figure 1 it shows a watermark embedding stage for watermarking relational databases. During this stage we use a key to determine the locations used to embed digital watermarks or produce digital watermarks in data distortion-free watermarking relational databases. In Figure 2, it shows a watermark retrieve stage for watermarking relational databases. During this stage, we also use the same key to find out the locations of watermarks. If we can't retrieve our watermark from a suspicious database, it means that this database is not the original database.

3.5 Basic Techniques

1) LSB (Least Significant Bit): It's the rightmost position in a binary integer, and can decide if the number is odd or even. Because it represents the smallest unit in a binary integer, i.e. the change of LSB of the number will be very small, it is usually used to hidden watermark information.

2) Data partition (Data grouping): It's a technique which can partition database into logical non-overlapping data partitions. The basic concept is that use a secret key, hash function and number of partitions to assign tuples to partitions [12]. Because these data partitions are logically partitioned, it won't separate physical data.

3) Majority voting: It's a voting rule in real life. When it is used in watermarking relational databases, its purpose is to correct decoded watermark bits [12]. For example, during decoding stage, if a watermark bit 1 in a data partition is over half the decoded bits, the decoded watermark bit of this data partition is 1.

4 Data Distortion Watermarking Relational Databases

As mentioned above, the techniques for watermarking relational databases are mainly grouped into two kinds [3, 6, 22]:

1) Data distortion watermarking relational databases: It directly embeds the digital watermark into some data in the relational database. This will make the data produce change, and these changes represent watermark information. However, the data distortion must be tolerable, or it will make the data become worthless.

2) Data distortion-free watermarking relational databases: Its main concept is that it first partitions data into several partitions, and then uses these partitions to generate the digital watermark. Because during a watermark embedding stage, it will not embed the watermark into the database, so it doesn't result in data distortion. The purpose of most of these techniques is to keep the integrity of data in relational databases because their generated watermarks are fragile.

In the data distortion watermarking relational databases, it has many schemes, such as image-based, speech-based, content-based, and others [8]. The papers we surveyed are AHK algorithms and other schemes of data distortion watermarking relational databases which are not mentioned in [8]. Besides AHK algorithms, these schemes are not in [8], in our opinion, they belong to others of data distortion watermarking relational databases. The papers we surveyed are as follows.

4.1 Agrawal-Kiernan's Scheme

The technique proposed by Rakesh Agrawal and Jerry Kiernan [1]. Their technique has two main phases:

1) Watermark insertion:
Watermark Insertion Algorithm is used to embed the

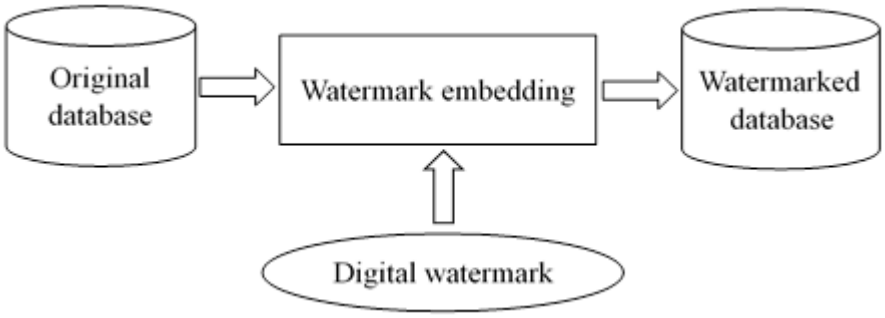


Figure 1: Watermark embedding stage for watermarking relational databases [16]

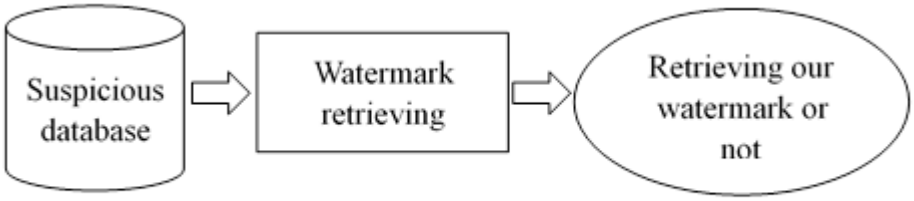


Figure 2: Watermark retrieve stage for watermarking relational databases [16]

watermark. This algorithm first uses hash function, primary key and private key e to mark one LSB of one numeric attribute of some tuples in the relational database, and then uses the value of the selected LSB to embed watermark bits: If the calculated value of the hash function (private key and primary key are passed as parameters) is even, then change the value of the selected LSB to 0; if it is odd, then change the value of the selected LSB to 1.

2) Watermark detection:
Watermark Detection Algorithm is used to retrieve the watermark. It first uses the same hash function, primary key and private key e to find out the LSBs which are embedded in the watermark. If the value of hash function is even (first hash is even) and the value of the selected LSB equals 0, then the watermark bit matches successfully. Similarly, if the value of hash function is odd and the value of the selected LSB equals 1, then it matches, too. The number of successful matches can determine whether the database is private or not.

Comment:

1) This technique is only suitable for numeric attributes. And it is assumed to modify the value of the LSB of numeric attributes and will not affect the usability of these data.

2) After a hacker changed schemes of relations, this technique would not find the original position embedded with a watermark. For example, adding or deleting an attribute in a relation, or changing the order of a relation [13].

3) Even if we can extract the complete watermark, because the extracted watermark don't have any ownership information, it is hard to clearly find whom the database belong to [13].

4) Their data distortion can be controlled arbitrarily by data owner through parameters: ν, ζ, and γ.

5) The requirement analysis is listed in Table 2.

Table 2: The requirement analysis of Agrawal-Kiernan's scheme [1]

Proposed Scheme	Agrawal et al. Technique
Robustness	Bit flipping, Mix-and-Match, Additive, & Invertibility attacks
Unambiguity	No
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.2 Sion-Atallah-Prabhakar's Scheme

The technique proposed by Radu Sion, Mikhail Atallah, and Sunil Prabhakar [26]. They proposed embedding

the watermark into data statistics. Their technique has two main phases, encoding phase and decoding phase. During encoding phase, it first partitions original data into subsets, and then uses Single Bit Encoding Algorithm to embed watermark bits into these subsets. During decoding phase, it first uses the partition technique of encoding phase to recover the subsets, and then uses Watermark Detection Algorithm to retrieve the watermark bits from these subsets. Finally, because these watermark bits may suffer from attacker's damage, it uses an error correcting mechanism to recover the most possible original watermark bits.

Comment:

1) Their proposed data partition technique is difficult to resist tuple deletion attack and tuple insertion attack [25].

2) During decoding phase, they use a threshold technique with two thresholds. However, they don't use optimal thresholds, and they pick thresholds at random instead [25].

3) Their data distortion can be controlled arbitrarily by data owner through data quality (goodness) metrics.

4) The requirement analysis is listed in Table 3.

Table 3: The requirement analysis of Sion-Atallah-Prabhakar's scheme [26]

Proposed Scheme	Sion et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks [25]
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.3 Shehab-Bertino-Ghafoor's Scheme

The technique proposed by Mohamed Shehab, Elisa Bertino, and Arif Ghafoor [25]. Their technique has two main phases:

1) Watermark encoding:

   a. Data set partitioning: Use a secret key Ks, number of partitions m and get_partitions algorithm to partition Data Set D into m non-overlapping data partitions $\{S_0, S_1, \cdots, S_{m-1}\}$.

   b. Watermark encoding: Use encode_single_bit algorithm to embed the watermark into partitions.

   c. Optimal threshold evaluation: Calculate the optimal threshold T^* to be used for decoding.

2) Watermark decoding:

   a. Data set partitioning: Use get_partitions algorithm of watermark encoding to find out partitions embedded with the watermark.

   b. Threshold-based decoding: Use optimal threshold T^* to decode watermark bits.It first computes the value of $\Theta(S_j, 0, c)$, and saves it into value. If value $\geq T^*$, it represents bit 1; else it represents bit 0.

   c. Majority voting: The watermark bit is determined through voting, and the majority of watermark bits are the final bit.

Comment:

1) This technique assumes that the tuples in every partition S_i all contain a numeric attribute, and therefore it is only suitable for numeric attributes.

2) Their data distortion can be controlled by data owner through usability constraints in G.

3) The requirement analysis is listed in Table 4.

Table 4: The requirement analysis of Shehab-Bertino-Ghafoor's scheme [25]

Proposed Scheme	Shehab et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

4.4 Kamran-Farooq's Scheme

The technique proposed by Kamran and Farooq [11]. Their technique has two main phases:

1) Watermark encoding:

   a. Data grouping: It first uses feature ranking to compute vector R_{nk} and C_{PT}, and then uses R_{nk}, C_{PT} and data grouping function to partition features into logical non-overlapping groups.

   b. Watermark embedding: Use Algorithm 1 to embed the watermark into non-numeric features. It first computes the hash value of each row, and then uses the order of these hash values to

embed watermark bits. These hash values will be saved in temp for decoding. Use Algorithm 2 to embed the watermark into numeric features of selected data groups. It uses the row value and $\triangle_i$ to embed watermark bits. If the row value adds positive $\triangle_i$, it represents a watermark bit 1; and if the row value adds negative $\triangle_i$, it represents a watermark bit 0.

2) Watermark decoding:

   a. Data grouping: Use data grouping function of Watermark encoding to find out the data groups embedded with the watermark.

   b. Watermark extraction: Use Algorithm 3 to extract the watermark from selected non-numeric features. It first gets hash values from temp, and then analyzes the order of these hash values. The descending order of hash values represents bit 1, and the ascending order of hash values represents bit 0. Use Algorithm 4 to extract the watermark from numeric features of selected data group. It uses decoding threshold T^* and a parameter val to decode. If $val > T^*$, it represents bit 1; else it represents bit 0.

Comment:

1) In Algorithms 1 and 3, the data stored in temp is too large. For example, temp needs to store the hash value of each row, if there are 10,000 rows, it needs to store 10,000 hash values.

2) This technique is not only suitable for numeric attributes, but also suitable for non-numeric attributes.

3) Their data distortion only happens in numeric attributes, and can be controlled through $\triangle_i$.

4) The requirement analysis is listed in Table 5.

Table 5: The requirement analysis of Kamran-Farooq's scheme [11]

Proposed Scheme	Kamran and Farooq Technique
Robustness	Alteration & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by $\triangle_i$

4.5 Kamran-Suhail-Farooq's Scheme

The technique proposed by Kamran, Suhail, and Farooq [12]. Their technique has three main phases:

1) Pick the position used to embed the digital watermark:

   a. Data Partitioning: Use a secret key Ks, number of partitions m and Algorithm 1 (Get_Partitions) to partition Data Set D into m non-overlapping data partitions $\{S,S_1,\cdots,S_{m-1}\}$.

   b. Selection of Data Set for Watermarking: Use Algorithm 2 (Get_Data_Selection_Threshold) to establish threshold for singling out the data sets from data partitions in (a).

   c. Hash Value Computation: Use Algorithm 3 (Get_Even_Hash_Value_Data Set) to decrease these data sets. And then we will get data sets which can be used to embed the watermark. By this way, the watermark is generated by Watermark Generating Function.

2) Watermark Embedding: Use Algorithm 4 (Embed_Watermark) to embed the watermark. It first computes the amount of data change. If a watermark bit is 1, the amount of data change is row value multiplied by positive ρ; and if a watermark bit is 0, the amount of data change is row value multiplied by negative ρ. And then it uses the row value plus the amount of data change to embed watermark bits.

3) Watermark Decoding: Use Algorithm 5 (Detect_Watermark) to detect the embedded watermark. This algorithm begins to detect watermarks after it uses Algorithms 1, 2 and 3 to find out data sets which are embedded with watermarks. Next step is to compute decoding threshold ν, and then use it to decode watermark bits. If $\nu \geq 0$, it represents bit 1; else it represents bit 0. Finally, the final watermark bits are determined through Majority voting.

Comment:

1) This technique is most suitable for unsigned numeric attributes.

2) Their data distortion can be controlled by data owner through ρ.

3) The requirement analysis is listed in Table 6.

4.6 Melkundi-Chandankhede's Scheme

The technique proposed by Swathi Melkundi and Chaitali Chandankhede [21]. Their technique has three main phases:

1) Watermark Insertion:

   a. Data Partitioning: Use a secret key Ks, number of partitions m and Algorithm 1 (Data Partition Algorithm) to partition Data Set D into m non-overlapping data partitions $\{P_0,P_1,\cdots,P_{m-1}\}$.

Table 6: The requirement analysis of Kamran-Suhail-Farooq's scheme [12]

Proposed Scheme	Kamran-Suhail-Farooq Technique
Robustness	Insertion, Alteration, Deletion, Multifaceted, Collusion, & Additive attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	Controlled by data owner

b. Insertion into a textual attribute: Use Unicode control characters ZWJ and ZWNJ to embed watermark bits. For ZWJ, its Unicode code point is U+200D and is the abbreviation of Zero width joiner. It is an invisible control character and used to represent a watermark bit 0. For ZWNJ, its Unicode code point is U+200C and is the abbreviation of Zero-width non-joiner. It is used to represent a watermark bit 1.

c. Insertion into numeric attribute: It first converts the value of the attribute into a binary value, and then flips the LSB of the binary value. That is, if you intend to embed a watermark bit = 0, then the LSB is changed to 0; if you intend to embed a watermark bit = 1, then the LSB is changed to 1.

2) Watermark Extraction:

a. Data Partitioning: Use Data Partition Algorithm of Watermark Insertion to find out the partitions embedded with the watermark.

b. Extraction from a textual attribute: If the value of the selected textual attribute in a data partition is ZWJ, it represents bit 0; and if the value is ZWNJ, it represents bit 1.

c. Extraction from numeric attribute: If the value of the LSB of the selected numeric attribute in a data partition is 0, it represents bit 0; and if the value is 1, it represents bit 1.

d. Majority voting: Through voting is used to determine the watermark bit, and the majority of watermark bits is the final bit.

3) Watermark Verification: Use Algorithm 2 (Watermark Verification Algorithm) to compare the extracted watermark with the raw watermark. Its concept is based on Levenshtein distance, and therefore it computes Levenshtein Distance between the extracted watermark and the raw watermark. If their difference is too large, it shows this database is not the original one.

Comment:

1) This technique is only suitable for a relation with numeric attributes and textual attributes at the same time.

2) Their data distortion only happens in numeric attributes, and the amount of data change is only the value of the LSB.

3) According to their description, their subset addition attack, subset deletion attack and subset alteration attack are our insertion attack, deletion attack and alteration attack, respectively.

4) The requirement analysis is listed in Table 7.

Table 7: The requirement analysis of Melkundi-Chandankhede's scheme [21]

Proposed Scheme	Melkundi et al. Technique
Robustness	Insertion, Alteration, & Deletion attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	LSB

4.7 Mehta-Pratap Rao's Scheme

The technique proposed by Brijesh B. Mehta and Udai Pratap Rao [20]. Their technique has three main phases:

1) Watermark insertion: It first uses hash function, primary key and private key k1 to select tuples in the database.

a. Insertion into a numeric attribute: Choose a LSB of a numeric attribute of selected tuples, and a watermark bit is substituted for the selected LSB.

b. Insertion into a date attribute: Choose seconds field (SS) of a date attribute of selected tuples, and embed watermark bits into the SS.

2) Watermark extraction: It first uses the same hash function, primary key and private key k1 to find out tuples which are embedded with the watermark.

   a. Extraction from a numeric attribute: Find out the LSB of the selected numeric attribute of these selected tuples, and the value of the LSB represents a watermark bit.

   b. Extraction from a date attribute: Find out the SS of the selected date attribute of these selected tuples, and extract watermark bits from the SS.

3) Watermark verification: Compare the extracted watermark with the raw watermark. It only needs the extracted watermark bits from one place instead of two places to match the original watermark bits successfully.

Comment:

1) This technique is only suitable for a relation with numeric attributes and date attributes at the same time. Because it actually embeds a watermark bit into two attributes (numeric, date) selected by k1 at the same time. Therefore, if this relation don't have the two attributes (numeric, date), it won't work.

2) Their data distortion happens in numeric attributes and date attributes. The amount of data change for numeric attributes is the value of the LSB, and the amount of data change for date attributes is SS.

3) According to their description, their subset addition attack, subset deletion attack, subset alteration attack and subset selection attack are our insertion attack, deletion attack, alteration attack and Mix-and-Match attack, respectively.

4) The requirement analysis is listed in Table 8.

Table 8: The requirement analysis of Mehta-Pratap Rao's scheme [20]

Proposed Scheme	Mehta et al. Technique
Robustness	Insertion, Alteration, Deletion, & Mix-and-Match attacks
Unambiguity	Yes
Security	Yes
Blindness	Yes
Imperceptibility (%)	100%
Usability	LSB and SS

1Because this technique uses data partition or data grouping technique, we think the watermark bits will distribute at random in the relational database.

5 Conclusion and Future Work

In this paper, we first introduce the history and background of watermarking relational databases, and then focus on surveying data distortion watermarking relational databases. Furthermore, we analyze these techniques by six requirements we mentioned in BACKGROURD. Next, we compare these techniques through requirement analysis table.

5.1 Comparison

Our comparison method is to rate them by scores. The best score is five points, and the worst score is one point. The result is as in Table 9.

About robustness, because Kamran and Farooq Technique only uses two basic attacks in their experiment, it scores 2 points. Kamran, Suhail and Farooq Technique can resist three basic attacks (Insertion attack, Alteration attack, Deletion attack) and three advanced attacks, so it scores the best grades.

About unambiguity, because Agrawal et al. Technique is hard to find any ownership information of the embedded digital watermark, it only scores 1 point. About security and blindness, every technique meets their conditions, and therefore these techniques all score 5 points. About imperceptibility, Agrawal et al. Technique, Sion et al. Technique, etc. scores 3 points because they only use basic data partition technique. Kamran, Suhail and Farooq Technique scores the highest grades because it uses advanced technique to further decrease the number of tuples which are ready to be watermarked, hence it has the best discontinuous degree of the watermark bits. About usability, because Melkundi et al. Technique's the amount of data distortion is only LSB, it scores 5 points.

According to total score, Kamran, Suhail and Farooq Technique is a better technique than others because it has a balanced performance in six requirements. It not only is the most robust technique, but also is the imperceptiblest. Therefore, we think a good technique for watermarking relational databases should consider all six requirements, it can't only focus on a few requirements.

5.2 Issues

Although watermarking relational databases has been developed over ten years, it still has some issues, and these issues are as follows:

1) Experiments: Unlike image processing domain, some scholars of watermarking relational databases usually perform their experiments with their own databases, and without comparing their technique with others in robustness, distortion, etc.. such as [28, 20, 9, 21]. Some scholars didn't perform experiment very well, for example, Javier et al. proposed a paper in 2014 [7]. Although they compared their technique with others in experiments, they still used their own database to perform experiments. Shehab et al.

Table 9: The comparison

Proposed Scheme	Robustness	Unambiguity	Security	Blindness	Imperceptibility	Usability	Total
Agrawal et al. Technique	4	1	5	5	3	3	21
Sion et al. Technique	3	5	5	5	3	3	24
Shehab et al. Technique	3	5	5	5	3	3	24
Kamran-Farooq Technique	2	5	5	5	4	3	24
Kamran-Suhail-Farooq Technique	5	5	5	5	5	3	28
Melkundi et al. Technique	3	5	5	5	3	5	26
Mehta et al. Technique	4	5	5	5	3	4	26

provide a good example in experiments in this domain [25].

They not only compare their technique with others, but also use an online database for their experiment. Therefore, after we read their research, we think that watermarking relational databases needs some open databases for everyone to do experiment. Therefore, we will provide a website that provides open data sets for everybody. Its internet address is: `http://archive.ics.uci.edu/ml/`. In watermarking relational databases, we strongly recommend that everyone should perform complete and fair experiments.

2) Data distortion: Although data distortion watermarking relational databases in academic research can tolerate data distortion, but for commercial purposes, data distortion in the relational database is not allowed. Even a bit of data distortion, it may cause a significant impact. Therefore, the commercial value of the research in the data distortion watermarking relational databases is not high, we believe that the goal of watermarking relational databases should develop towards data distortion-free watermarking relational databases or reversible watermarking relational databases, and data distortion watermarking relational databases should be eliminated.

5.3 Future Work

1) As mentioned above in B.2), for the purpose of distortion-free data, we can consider not to embed the watermark into the content of a database, but other places of a database, such as the comment of a table, database relationship [23], or using the number of tables in a large database to embed the watermark, etc. Because we don't embed digital watermarks into the database, we don't damage the raw data, it can achieve the purpose of distortion-free data.

2) Computation time: As far as we know, most of watermarking relational databases don't consider computation time in experiments except [7]. We think that computation time is another important issue except robustness. Because in the age of big data, the amount of data will become bigger and bigger, and then the amount of data will affect the computation time. A technique which takes a lot of computation time is worthless. Therefore, in addition to robustness, computation time should be considered in experiments. In the future, we must strike a balance between robustness and computation time.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant MOST 104-2221-E-468-004. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Agrawal and J. Kiernan, "Watermarking relational databases", in *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 155–166, Hong Kong, China, 2002.

[2] A. Al-Haj and A. Odeh, "Robust and blind watermarking of relational database systems", *Journal of Computer Science*, vol. 4, no. 12, pp. 1024–1029, 2008.

[3] M. H. Bhesaniya, J. Rathod, and K. Thanki, "Various approaches for watermarking of relational database", *International Journal of Engineering Science and Innovative Technology*, vol. 3, no. 1, pp. 215–220, 2014.

[4] M. H. Bhesaniya, K. Thanki, "Watermarking of relational databases", *International Journal for Research in Technological Studies*, vol. 1, no. 1, pp. 11–16, 2013.

[5] L. Camara, J. Li, R. Li, and W. Xie, "Distortion-free watermarking approach for relational database integrity checking", *Mathematical Problems in Engineering*, Article ID 697165, 2014.

[6] A. K. Dwivedi, B. K. Sharma, and A. K. Vyas, "Watermarking techniques for ownership protection of re-

lational databases", *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 1, pp. 368–375, 2014.

[7] J. Franco-Contreras, G. Coatrieux, F. Cuppens, N. Cuppens-Boulahia, C. Roux, "Robust lossless watermarking of relational databases based on circular histogram modulation", *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 397–410, 2014.

[8] R. Halder, S. Pal, and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison", *Journal of Universal Computer Science*, vol. 16, no. 21, pp. 3164–3190, 2010.

[9] D. Hanyurwimfura, L. Yuling, and L. Zhijie, "Text format based relational database watermarking for non-numeric data", in *International Conference On Computer Design And Appliations (ICCDA'10)*, vol. 4, pp. 312–316, 2010.

[10] S. Iftikhar, M. Kamran, and Z. Anwar, "RRW-A robust and reversible watermarking technique for relational data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1132–1145, 2015.

[11] M. Kamran and M. Farooq, "A formal usability constraints model for watermarking of outsourced datasets", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 1061–1072, 2013.

[12] M. Kamran, S. Suhail, and M. Farooq, "A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints", *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2694–2707, 2013.

[13] C. H. Ke, M. S. Wang, *A Study of Watermarking in Relational Database*, M.S. Thesis, Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan, 2006.

[14] S. Khanna and F. Zane, "Watermarking maps: Hiding information in structured data", in *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 596–605, San Francisco, California, USA, 2000.

[15] S. S. Kshatriya and S. S. Sane, "A Study of Watermarking Relational Databases", *International Journal of Application or Innovation in Engineering & Management*, vol. 3, no. 10, pp. 154–158, 2014.

[16] Y. Li, "Database watermarking: A systematic view", in *Handbook of Database Security*, pp. 329–355, Springer US, 2008.

[17] Y. Li, H. Guo, and S. Jajodia, "Tamper detection and localization for categorical data using fragile watermarks", in *Proceedings of the 4th ACM Workshop on Digital Rights Management*, pp. 73–82, Washington DC, USA, 2004.

[18] Y. Li, V. Swarup, and S. Jajodia, "Constructing a virtual primary key for fingerprinting relational data", in *Proceedings of the 3rd ACM Workshop on Digital Rights Management*, pp. 133–141, Washington, DC, USA, 2003.

[19] B. B. Mehta and H. D. Aswar, "Watermarking for security in database: A review", in *IEEE Conference on IT in Business, Industry and Government (CSIBIG'14)*, pp. 1–6, 2014.

[20] B. B. Mehta and U. P. Rao, "A novel approach as multi-place watermarking for security in databas", in *International Conference on Security and Management*, pp. 703–707, 2011.

[21] S. Melkundi and C. Chandankhede, "A robust technique for relational database watermarking and verification", in *IEEE International Conference on Communication, Information & Computing Technology (ICCICT'15)*, pp. 1–7, 2015.

[22] A. A. Mohanpurkar and M. S. Joshi, "Applying watermarking for copyright protection, traitor identification and joint ownership: A review", in *IEEE World Congress on Information and Communication Technologies (WICT'11)*, pp. 1014–1019, 2011.

[23] H. Pieterse and M. Olivier, "Data hiding techniques for database environments", in *IFIP Advances in Information and Communication Technology*, Advances in Digital Forensics VIII, pp. 289–301, Springer Berlin Heidelberg, 2012.

[24] V. Prasannakumari, "A robust tamperproof watermarking for data integrity in relational databases", *Research Journal of Information Technology*, vol. 1, no. 3, pp. 115–121, 2009.

[25] M. Shehab, E. Bertino, and A. Ghafoor, "Watermarking relational databases using optimization-based techniques", *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 116–129, 2008.

[26] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for relational data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 1509–1525, June 2004.

[27] M. H. Tsai, H. Y. Tseng, and C. Y. Lai, "A Database Watermarking Technique for Temper Detection", in *Proceedings of the 2006 Joint Conference on Information Sciences (JCIS'06)*, Kaohsiung, Taiwan, Atlantis Press, 2006.

[28] Y. Y. Yang, D. C. Wu, W. H. Tsai, "Watermarking of numerical databases using spread spectrum techniques", in *Proceedings of 4th Workshop on Digital Archives Technologies*, pp. 79-84, 2005.

[29] Y. Zhang, B. Yang, and X. M. Niu, "Reversible watermarking for relational database authentication", *Journal of Computers*, vol. 17, no. 2, pp. 59–66, 2006.

Ming-Ru Xie received the B.S. in Computer and Information Science from Aletheia University, New Taipei City, Taiwan, Republic of China, in 2002. He had worked in IT industry in Taiwan for ten years. He is currently a master's degree student in the Department of Management Information System, National Chung Hsing University, Taichung, Taiwan. His current research interests include database security, information security, and digital image techniques.

Chia-Chun Wu received a Ph.D. degree in Department of Computer Science and Engineering from National Chung-Hsing University, Taichung, Taiwan, in 2011. He is currently an assistant professor at the Department of Industrial Engineering and Management, National Quemoy University, Kinmen County, Taiwan. His current research interests include database security, secret image sharing, mobile applications development, and digital image techniques.

Jau-Ji Shen received his Ph.D. degree in Computer Science and Information Engineering in 1988 from National Taiwan University, Taipei, Taiwan. Currently, he is a professor at Management Information Systems, National Chung Hsing University, Taichung, Taiwan. His research interests include software quality assurance, data and knowledge techniques, and digital image techniques.

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.

Discovering Cyber Terrorism Using Trace Pattern

Nurhashikin Mohd Salleh, Siti Rahayu Selamat, Robiah Yusof, and Shahrin Sahib
(Corresponding author: Nurhashikin Mohd Salleh)

Faculty of Information Technology and Communication, Universiti Teknikal Malaysia Melaka (UTeM)
(Email: nurhashikinbtmohdsalleh92@gmail.com)
(Received Aug. 13, 2015; revised and accepted Nov. 27, 2015)

Abstract

Nowadays, as the Internet user increased, the number of cyber threats is also increased. Internet has provided a medium for criminal to do the crime and become the target for cyber terrorist to spread their negative propaganda, and promote extreme activities. One of the crimes is cyber terrorism. Cyber terrorism became more sophisticated and it difficult to discover its activities. Hence, this paper proposes tracing technique for discovering cyber terrorism based on trace pattern. Trace pattern will represent the behavior and activities of cyber terrorism. Cyber terrorist's website is used as the datasets. Using tracing technique, cyber terrorist's activities are identified by extraction and classifying the traces to the keyword that is usually used by the terrorist. Then, the traces will be linked with the cyber terrorism components in order to identify the relationship between them. Using trace pattern, the verification process will be conducted to verify the traces in order to identify the cyber terrorism activities and potential terrorist. This trace pattern can be used in facilitating the forensic investigation process in discovering cyber terrorism activities.

Keywords: Critical national information infrastructure (CNII), cyber terrorism, trace pattern, tracing technique

1 Introduction

Cyber terrorism became more sophisticated and it difficult to discover its activities. Since, Internet provides the platform to the user, cyber terrorist could take advantage of the Internet and other IT infrastructure as their target to do terrorist activities. Cyber terrorists may use Internet as the medium for hacking, spreading negative propaganda, and promoting extreme activities. They may also use Internet for the purpose of inter-group communication and inter-networked grouping [15] and create public relations [8] to spread their propaganda and promote their extremist activities. It became more extreme when Critical National Information Infrastructure (CNII) became an attractive target to the cyber terrorist. Cyber terrorism might be attack against information, computer system, computer programs, or data which result in violence [1]. As the result, it could leave the nation with difficult conditions due to the disruption of critical services. It became more difficult to discover its activities.

Therefore, the main objective of this paper is to propose a tracing technique to discover cyber terrorism using trace pattern in facilitating the investigator on identifying cyber terrorist activities and provide the evidence. The traces of the crime are discovered by using tracing technique in order to formulate trace pattern. The potential terrorist website is used.

2 Related Work

2.1 Overview of Cyber Terrorism

Cyber terrorism is the use of internet to launch any terrorist attacks or threats through computer system, computer program, or data which result on damage critical infrastructure or at least cause harm and generate fear to other target. Cyber terrorism continues to rise, and terrorists increase in a cyber space [2]. Terrorists use the Internet as a tool to coordinate action, intra-group communications, fund-raising and public relations [7, 8, 9]. For example, terrorist organize websites for hacking, spreading negative propaganda, and promoting extreme activities. It is very difficult to discover cyber terrorism from carrying out their activities since there is no specific trace pattern [11]. Based on [14], there are at least six components required to describe cyber terrorism activities as shown in Figure 1.

Figure 1 shows the cyber terrorism components consists of actor, motivation, tools, target, method, and impact. Actor can be described as a participant in any action or process. It refers to any person, group, or organization. Motivation refers to any reason of acting or behavior in a particular way. It can be any concept, ideology, or revenge. Tool is a device used to carry out a particular action like weapon or network warfare. Target refers to person, organization, government, society, objects, or

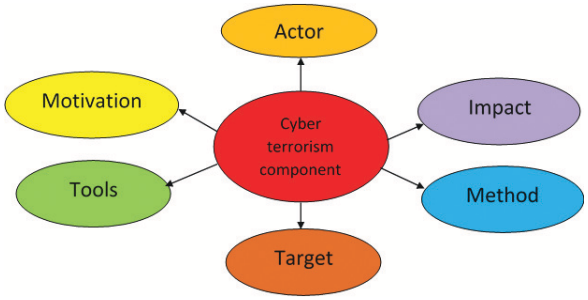


Figure 1: Cyber terrorism components

place that selected as the aim of an attack. Method is a particular procedure for accomplishing or approaching something. Method refers to the any action or operation that relate to the cyber terrorism. Impact defines as a marked effect or influence, and can be classified into four categories namely physical, psychology, social, and economic. Every violence and threats that has been done to the target will be considered as impact [3, 4, 5, 13]. Cyber terrorism components can be represents in terms of "union" statements equation as in Equation (1).

$$CcT = c_1 \cup c_2 \cup c_3 \cup c_4 \cup c_5 \cup c_6 \quad (1)$$

Where, CcT = cyber terrorism components, c_1 = actor components, c_2 = motivation components, c_3 = tools components, c_4 = target components, c_5 = method components, and c_6 = impact components. Equation (1) shows the components of cyber terrorism either actor, motivation, tools, target, method, or impact components.

2.2 Tracing Technique

Tracing technique can be defined as a process of tracing the data based on its attribute [11]. In this research, tracing technique is used to help the forensic investigator to trace any cyber terrorism activities based on its components. By using this technique, behavior of cyber terrorism can be identified.

Two main tracing techniques used namely word searching technique and web address searching techniques. Word searching technique is use to trace the data. This technique can be use by using symbols to search for alternative word endings, combining the concepts in a search statement, searching for phrases, and performing more specific searches. By using this technique, the project can be identified what the keyword and its type. Keyword refer to any word exists in website. The examples of these types are people, organization, group, individual, target place, place, action, mode of action, tools, and motivation [6].

2.3 Trace Pattern

Trace pattern can be defined as the way to discover the origin or starting point of a scenario that has happened [10]. It plays an important role by representing the behavior and activities of cyber terrorism. With trace pattern, it will assist forensic investigators on tracing the traces left at the crime scenes and discovering cyber terrorism activities. Trace pattern will help the forensic investigator to find any evidence about the cyber terrorism because any activities of cyber terrorists or attacker can be identified based on the traces data found in the attack pattern which represent in the form of trace pattern. In this situation, trace pattern will help to determine how cyber terrorism could be happened [12].

In this research, the traces can be any keywords appeared in the terrorist website. Therefore, in order to get a trace pattern, keywords of these traces is identifying based on its meaning and its relation to cyber terrorism components. After that, the traces will be confirming its relation.

3 Methodology

There are three phases involved in the tracing technique for discovering cyber terrorism activities based on the trace pattern. The explanations of these phases are explained below.

Phase I: Traces extraction and classification
The tracing technique namely word searching technique and web searching techniques are used to extract the traces. In order to formulate trace pattern, these technique are used to discover the traces. Traces are referred to any keyword used by the terrorist. The keyword can be word or URL. In this phase, the traces are classified into types of keyword as described as in Table 1.

Table 1 shows the types of keyword which are person, group, organization, concept, ideology, critical infrastructure, action, operation, physical, psychological, emotional, and economic. The process of classifying the keyword traced to the types of keyword based on the meaning of the word. For example, the name of the person is one of the words that referred to an individual and based on Table 1, this trace is classified as person.

Phase II: Traces cross-referencing and linking
The main purpose of this process in this phase is obtaining complete traces of cyber terrorism activities. Therefore, after the traces classified in the Phase I, the traces will be linked in order to identify the relationship between them. The cross-referencing is also carried out to identify the relationship between the traces and the cyber terrorism components.

Phase III: Traces verification
In this phase, the traces that are linked in the Phase II will be verified to identify the cyber terrorism activities and indirectly identifying the potential terrorist or the suspect. The verification is used the trace pattern. Trace pattern will describe about the types of keyword and the

Table 1: Types of keyword

Types of keyword	Description
Person	An individual
Group	A number of people or things that are located, gathered, or classed together.
Organization	An organized group of people with a particular purpose, such as a business or government department.
Concept	An idea of something is and how it works.
Ideology	A system of ideas and ideals, especially one which forms the basis of economic or political theory and policy. It also known as the set of beliefs characteristic of a social group or individual
Critical infrastructure	The backbone of nation's economy, security and health. It also known as the assets, systems, and networks, whether physical or virtual.
Action	The fact or process of doing something, typically to achieve an aim.
Operation	It known as any job or tasks consisting of one or more elements or subtasks.
Physical	Involve body contact.
Psychological	Relate to the mental and emotional of a person.

cyber terrorism components involved. For every types of keyword identified in the website are mapped with the cyber terrorism components. The components could be actor, motivation, tools, method, target, and impact. The mapping process is shown in Figure 2.

Figure 2 shows cyber terrorism trace pattern that used in this research as the tracing technique to discover the traces and identify the cyber terrorism. Trace pattern describes the activities of terrorist on how terrorist attack the target and others. This process can be represents in terms of if then statements equation as in Equation (2).

$$IF(a_n == TK_n)THEN(TK_n == CcT_n) \quad (2)$$

Where, a = attributes, TK = types of keyword, CcT = cyber terrorism components, and n = number. Equation (2) is used to verified the traces and discover cyber terrorism activities using trace pattern.

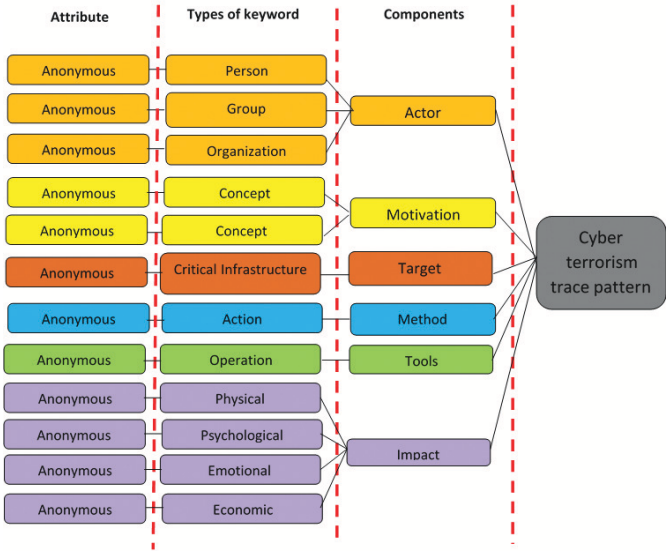


Figure 2: Cyber terrorism trace pattern

4 Proposed Tracing Technique for Discovering Cyber Terrorism Based on Trace Pattern

From analysis and findings, by using trace pattern it able to discover cyber terrorism activities. We proposed tracing technique for discovering cyber terrorism based on the trace pattern as shown in Algorithm 1, Algorithm 2 and Algorithm 3.

4.1 Extraction and Classification

In this process, the traces of the cyber terrorism are extracted and classified into the types of keyword. The algorithm of the extraction and classification process is depicted in Algorithm 1.

Algorithm 1 Extraction and classification data algorithm

```
Start
Read website
Identify traces
if traces == word then
    traces = types of keyword
else if traces == URL then
    traces = types of website
else
    Identify the traces again
end if
End
```

Algorithm 1 shows the pseudo code of extraction and classification data. In this algorithm, the function will read website and identify the traces whether it is word or URL. If the traces equal to word, then traces will be types of keyword. Meanwhile, if traces equal to URL,

then traces will be types of website. If not, the function will be identifying the traces again.

4.2 Cross-referencing and Linking

The traces found in the extraction and classification process are then linked. The process is shown in Algorithm 2.

Algorithm 2 Extraction and classification data algorithm

```
Start
Read website
Identify attribute
if (types of keyword == characteristics) AND characteristics == components) then
    Map the keyword with components
else if (types of website == characteristics) AND (characteristics == components) then
    Map the keyword with components
else
    Identify the attribute again
end if
End
```

Algorithm 2 shows the pseudo code of cross-referencing and linking process. In this algorithm, the function will be read website and identify the attribute whether it is types of keyword or types of website. If the types of keyword is similar to the characteristics and characteristic is equal to components of cyber terrorism, the cross-referencing is then done to the types of keyword and cyber terrorism components. Then, these traces are linked to identify the relationship between them. If the type of website is same to the characteristics and characteristic is equal to components of cyber terrorism, then map the types with its components. If not, the function will be identifying the attribute again.

4.3 Cyber Terrorism Identification

After all relevance traces extracted, classified and linked, there is a need to verify the traces in order to identify the cyber terrorism activities by comparing to the cyber terrorism pattern as described in previous section. The algorithm of identification process is depicted in Algorithm 3.

Algorithm 3 depicts the pseudo code of cyber terrorism identification process. If the type of website is same to the characteristics and characteristic is equal to components of cyber terrorism, then map the types with its components. If not, the function will be identifying the attribute again. Then, after finished tracing the traces, the traces are then compared with the trace pattern in order to verified the cyber terrorism activities and identifying the potential terrorist.

Algorithm 3 Identification algorithm

```
Start
Read Traces
Traces = keyword OR Traces = website
if (types of keyword == characteristics) AND characteristics == components) then
    Map the keyword with components
else if (types of website == characteristics) AND (characteristics == components) then
    Map the keyword with components
else
    Identify traces again EOF
end if
Compare traces
End
```

5 Analysis and Findings

In this section, the ability of tracing technique for discovering cyber terrorism activities based on the trace pattern is identified. The capabilities of tracing technique need to be measured to identify its effectiveness to discover the traces based on the total percentage of relevant traces discovered in the website. Thus, the metric used in this research is known as Tracing Percentage (TC_p) used to measure the effectiveness of tracing process. Tracing Percentage for each component (TC_n) is the ratio of related traces discovered ($N_{related\_traces}$) and the total traces (N_{total_traces}) expressed in percentage value. These metric is represented as Equations (3) and (4).

$$TC_n = \frac{N_{related\_traces}}{N_{total\_traces}} \times 100 \quad (3)$$

$$TC_p = TC_n + TC_{n+1} + TC_{n+2} + ... + TC_{n+i}. \quad (4)$$

Equation (3) shows the equation to calculate traces percentage for each component while Equation (4) is the equation to calculate total percentage of the components discovered. In this research, three dataset are analyzed to identify the ability of tracing technique to discover cyber terrorism activities based on the trace pattern. These dataset (DS) are describes in Table 2.

Table 2: Dataset description

Dataset (DS)	Description
DS1	Website name: Anonymous W1, Website type: Blog, URL: Anonymous U1
DS2	Website name: Anonymous W2, Website type: Blog, URL: Anonymous U2
DS3	Website name: Anonymous W3, Website type: Blog, URL: Anonymous U3

Table 3: Total percentage of related traces

DS	Components	N_{total_traces}	$N_{related\_traces}$	TC_n %
DS1	Actor	153	57	37.25
	Motivation	153	55	35.95
	Tools	153	5	3.27
	Method	153	17	11.11
	Target	153	5	3.27
	Impact	153	6	3.92
			TCp = 94.77 %	
DS2	Actor	309	186	60.19
	Motivation	309	36	11.65
	Tools	309	7	2.27
	Method	309	24	7.77
	Target	309	48	15.53
			TCp = 97.41 %	
DS3	Actor	272	127	46.69
	Motivation	272	41	15.07
	Tools	272	12	4.41
	Method	272	31	11.40
	Target	272	40	14.71
			TCp = 92.28 %	

Table 2 shows the information of dataset using namely DS1, DS2, DS3. The names of all websites are hidden due to the sensitive issues. The scripting is use to trace and identify the frequent of the keyword which contribute to the cyber terrorism activities. By using Equations (2) and (3), the tracing percentage for three DS is shown in Table 3.

Table 3 shows the total percentage of related traces for DS. It consists of actor, motivation, tools, method, target, and impact components. The result obtained in Table 3 demonstrates the Tracing Percentage (TC_p) is able to discover the cyber terrorism activities by identifying the components and the types of keyword. These abilities are demonstrated through the result obtained using three DS; DS1, DS2, and DS3. For each DS, the total traces found for each types of keyword and components are shown in Tables 4, 5, and 6 respectively. Due to the sensitive data, the keyword traced in this paper is represented as K_1 to K_n.

Tables 4, 5, and 6 shows the traces found in DS1, DS2, and DS3 respectively. Scripting is used in the analysis and design process to trace the frequent keyword appears that identified. Based on these tables, the components of cyber terrorism have appeared in all DS except one of cyber terrorism components known as impact is not found in DS2 and DS3. However, this component is found in DS1 with the keyword identified is Threat. From the traces found as shown in all tables, this research identified that Actor can be person, group, organization, or place. While, by having any concepts that make a person become motivated or fanatic, it can contribute to cyber terrorism activities which classified as the Motivation component. This was proved that most of the traces found is belong to this component. This can be proved when the traces found in the three DS are majority below to motivation components. For example, there are 55 traces found on keyword concept in DS1, 36 traces in DS2, and 41 traces in DS3. The findings also shown that, in all attacks, target become the important component in cyber terrorism activities in which it was found in all DS analyzed.

6 Conclusion and Future work

This paper introduces the tracing technique for discovering cyber terrorism activities based on the trace pattern. Two main tracing techniques used namely word searching technique and web address searching techniques. Word searching technique is use to extract the traces. Tracing technique consists of traces extraction and classification, traces cross-referencing and linking, and traces verification. Tracing Percentage (TC_P) used to measure the effectiveness and the capabilities of tracing technique to discover the traces based on the total percentage of relevant traces discovered in the website. The abilities of tracing technique are demonstrated through the result obtained using DS. From the research findings, we are able to derive the equation to represent the components of the cyber terrorism.

Acknowledgments

The authors would like to thank INSFORNET Research Group of Universiti Teknikal Malaysia Melaka

Table 4: Traces found in DS1

Keyword	Count	Types of keyword	Components
K_1	27	Concept	Motivation
K_2	21	Concept	Motivation
K_3	18	Person	Actor
K_4	10	Group	Actor
K_5	9	Organization	Actor
K_6	8	Mode of action	Method
K_7	7	Concept	Motivation
K_8	6	Person	Actor
K_9	6	Threat	Impact
K_{10}	5	Tools	Tools
K_{11}	5	Target place	Target
K_{12}	5	Mode of action	Method
K_{13}	4	Person	Actor
K_{14}	4	Person	Actor
K_{15}	4	Mode of action	Method

Table 5: Traces found in DS2

Keyword	Count	Types of keyword	Components
K_1	48	Target Place	Target
K_2	43	Group	Actor
K_3	35	Group	Actor
K_4	30	Concept	Motivation
K_5	27	Organization	Actor
K_6	20	Group	Actor
K_7	18	Mode of action	Method
K_8	17	Place	Actor
K_9	17	Group	Actor
K_{10}	11	Person	Actor
K_{11}	10	Place	Actor
K_{12}	7	Tools	Tools
K_{13}	6	Concept	Motivation
K_{14}	6	Person	Actor
K_{15}	6	Action	Method

Table 6: Traces found in DS3

Keyword	Count	Types of keyword	Components
K_1	40	Target place	Target
K_2	35	Concept	Motivation
K_3	31	Group	Actor
K_4	31	Group	Actor
K_5	25	Organization	Actor
K_6	23	Mode of action	Method
K_7	12	Person	Actor
K_8	12	Tools	Tools
K_9	11	Place	Actor
K_{10}	10	Place	Actor
K_{11}	8	Action	Method
K_{12}	7	Person	Actor
K_{13}	6	Concept	Motivation

(UTeM) for the financial support under the Fundamental Research Grant Scheme with Project No. FRGS/1/2015/ICT04/UTeM/02/5.

References

[1] M. Dawson and M. Omar, *New Threats and Countermeasures in Digital Crime and Cyber Terrorism*, Information Science Reference, 2015.

[2] S. Gilmour, "Policing crime and terrorism in cyberspace: An overview," *The European Review of Organised Crime*, vol. 1, no. 1, pp. 143–159, 2014.

[3] S. Gordon and R. Ford, "Cyberterrorism?," *Computers & Security*, vol. 21, no. 7, pp. 636–647, 2002.

[4] S. Gordon and R. Ford, "On the definition and classification of cybercrime," *Journal in Computer Virology*, vol. 2, no. 1, pp. 13–20, 2006.

[5] R. Heickerö, "Cyber terrorism: Electronic jihad," *Strategic Analysis*, vol. 38, no. 4, pp. 554–565, 2014.

[6] L. Jarvis and S. Macdonald, "What is cyberterrorism findings from a survey of researchers," *Terrorism and Political Violence*, vol. 27, no. 4, pp. 657–678, 2015.

[7] M. Kenney, "Cyber-terrorism in a post-stuxnet world," *Orbis*, vol. 1, no. 59, pp. 111–128, 2015.

[8] J. A. Lewis, "Assessing the risks of cyber terrorism, cyber war, and other cyber threats," Center for Strategic and International Studies, Nov. 2002.

[9] O. Oluwafemi, F. A. Adesuyi, and S. M. Abdulhamid, "Combating terrorism with cybersecurity: The nigerian perspective," *World Journal of Computer Application and Technology*, vol. 1, no. 4, pp. 103–109, 2013.

[10] S. R. Selamat, R. Yusof, S. Sahib, M. Z. Masu'd, M. F. Abdollah, and Z. Z. Abidin, "Advanced trace pattern for computer intrusion discovery," *Journal of Computing*, vol. 2, no. 6, pp. 200–207, 2010.

[11] S. R. Selamat, N. M. Salleh, R. Yusof, and S. Sahib, "Constructing cyber terrorism trace pattern for forensic investigation process," in *Proceedings of the 14th International Conference on Applied Computer and Applied Computational Science*, Recent Advances in Computer Science, pp. 240–245, 2015.

[12] S. R. Selamat, R. Yusof, S. Sahib, N. F. Hassan, M. F. Abdollah, and Z. Z. Abidin, "Traceability in digital forensic investigation process," in *IEEE Conference on Open Systems*, pp. 101–106, Sept. 2011.

[13] R. Smith, "The cyber terrorism threat to critical infrastructure," Master thesis, Utica College, 2014.

[14] Z. Yunos, "Cyber terrorism conceptual framework," in *Proceeding in OIC-Certificate Annual Conference*, pp. 1–8, 2012.

[15] Z. Yunos and R. Ahmad, "Evaluating cyber terrorism components in malaysia," in *The 5th International Conference on Information and Communication Technology for The Muslim World*, pp. 1–6, Nov. 2014.

Nurhashikin Mohd Salleh is currently a Master student at the Universiti Teknikal Malaysia Melaka, Malaysia. She holds Bachelor of Computer Science (Hons) in Computer Networking. Her research interests include network forensic, cyber terrorism and cyber violent.

Siti Rahayu Selamat is currently a lecturer at the Universiti Teknikal Malaysia Melaka, Malaysia. She received her Doctor of Philosophy in Computer Science. Her research interests include network forensic, cyber terrorism, intrusion detection, network security and penetration testing.

Robiah Yusof is currently a lecturer at the Universiti Teknikal Malaysia Melaka, Malaysia. She received her Doctor of Philosophy in Computer Science. Her research interests include network management, network forensic, intrusion detection, network security and malware.

Shahrin Sahib received the Bachelor of Science in Engineering, Computer Systems and Master of Science in Engineering, System Software in Purdue University in 1989 and 1991 respectively. He received the Doctor of Philosophy, Parallel Processing from University of Sheffield in 1995. He is a professor and the Vice Chancellor of Universiti Teknikal Malaysia Melaka. His research interests include network security, computer system security, network administration and network design. He is a member panel of Experts National ICT Security and Emergency Response Center and also Member of Technical Working Group: Policy and Implementation Plan, National Open Source Policy.

Security Extension for Relaxed Trust Requirement in Non3GPP Access to the EPS

Hiten Choudhury[1], Basav Roychoudhury[2] and Dilip Kr. Saikia[3]
(Corresponding author: Hiten Choudhury)
Department of Computer Science & IT, Cotton College State University[1]
Panbazar, Guwahati-1, Assam, India
Indian Institute of Management, Meghalaya, Shillong-14, India[2]
Department of Computer Science and Engineering, National Institute of Technology Meghalaya[3]
Laitumkhrah, Shillong-3, India
(Email: hiten.choudhury@gmail.com)
(Received Sept. 12, 2015; revised and accepted Dec. 7, 2015)

Abstract

Third Generation Partnership Project (3GPP) has standardized the Evolved Packet System (EPS) as a part of their Long Term Evolution System Architecture Evolution (LTE/SAE) initiative. In order to provide ubiquitous services to the subscribers and to facilitate interoperability, EPS supports multiple access technologies where both 3GPP and Non-3GPP defined access networks are allowed to connect to a common All-IP core network called the Evolved Packet Core (EPC). However, a factor that continues to limit this endeavor is the trust requirement with respect to the subscriber's identity privacy. There are occasions during Non-3GPP access to the EPS when intermediary network elements like the access networks that may even belong to third party operators have to be confided with the subscriber's permanent identity. In this paper, we propose a security extension that relaxes this need. Contrary to several other solutions proposed recently in this area, our solution can be adopted as an extension to the existing security mechanism.

Keywords: EPS, EPC, non 3GPP access, privacy, trust

1 Introduction

3GPP has standardized the EPS, as a part of their LTE/SAE initiative. EPS supports multiple access technologies, through a common All-IP core network called the Evolved Packet Core (EPC).

In order to expand the reach of 3GPP services, 3GPP has proposed TS 23.402 [5]. This specification specifies description for providing IP connectivity using Non-3GPP Access Networks (Non-3GPP ANs) like WiMAX, WLAN [22], etc., to the EPC.

Non-3GPP access can be split into two categories viz., trusted and untrusted. For trusted Non-3GPP access, the subscriber's User Equipment (UE) connects directly with the EPC through the Non-3GPP AN. Whereas, for untrusted Non-3GPP access, an Internet Protocol Security (IPsec) tunnel is established between the UE and the EPC [4]. The tunnel provides end to end confidentiality between the UE and the EPC, thereby relaxing the need for the subscriber and the EPC to trust the untrusted Non-3GPP AN with signalling/user data exchanged through it. Such trust relaxation facilitates interoperability, as it simplifies agreements between the 3GPP and the Non-3GPP operators. However, a factor that continues to limit interoperability is the trust requirement with respect to the subscriber's identity privacy.

Each UE is assigned a unique and a permanent identity called the International Mobile Subscriber Identity $IMSI$. This identity is assigned by the 3GPP service provider so that the UE may be uniquely identified. The $IMSI$ is a precious information that needs to be protected. Knowledge of the $IMSI$ of a subscriber may allow an adversary to track and amass comprehensive profiles about subscribers. Such profiling may expose an individual to various kind of risks, and above all may deprive an individual of his privacy. Thus, knowledge of the $IMSI$ should be restricted to the UE and its home network.

In the Authentication and Key Agreement (AKA) protocol used to provide access security in Non-3GPP access to the EPS, there are occasions when intermediary network elements like the Non-3GPP AN has to be confided with the $IMSI$ of the subscriber through the vulnerable radio link. Such trust requirement not only limits interoperability by complicating agreements/pacts between 3GPP and Non-3GPP operators, but also provides scope for eavesdroppers to compromise the $IMSI$. In today's context when multiple operators collaborate with each other to provide wider coverage, such trust requirement imposes restriction and adds overhead in providing ubiq-

uitous service to the subscribers.

In this paper, we propose a security extension for the AKA protocol used to provide access security in Non-3GPP access to EPS. The extension follows an end to end approach, where the knowledge of the $IMSI$ is restricted only to the UE and its home network, thereby relaxing the need to trust intermediary network elements like the Non-3GPP AN with the $IMSI$. Thus, the extension not only enhances identity privacy of the subscribers but also helps in setting up a conducive platform for flexible on-demand and on-the-fly agreements between the EPS and the Non-3GPP AN, instead of complicated prior agreements/pacts (with complicated trust requirements). Unlike several other solutions proposed in this area, the main strength of our proposal is that it can be adopted as an extension to the existing security mechanism. Moreover, it has to be implemented only at the operators level without tasking the intermediary network elements (that may even belong to third party operators). Hence, making it easier for an operator that already has numerous subscribers registered with it, to adopt this extension.

The rest of the paper is organized as follows: in Section 2, we present a simplified view of the security architecture for Non-3GPP access to the EPS; in Section 3, we discuss access security and the status of identity privacy in Non-3GPP access to the EPS; in Section 4, we review the literature for some of the related work done in this area; in Section 5, we put forward our security extension for the AKA protocol used during Non-3GPP access to the EPS; in Section 6, we perform a formal analysis of the proposed extension to prove that it meets its security goals; from Section 7 through Section 9, we perform computation, space and communication overhead analysis of the proposed security extension; finally in Section 10, we conclude the paper.

2 Security Architecture of Non-3GPP Access to the EPS

Figure 1, depicts a simplified view of the security architecture for Non-3GPP access to the EPS. The 3GPP Authentication Authorisation Accounting Server (3GPP AAA Server) is located at the Home Public Land Mobile Network (HPLMN). Its primary responsibility is to authenticate the subscriber, based on authentication information retrieved from the Home Subscription Server (HSS). The authentication signalling may pass via several AAA Proxies. The AAA Proxies that are used to relay AAA information may reside in any network between the Non-3GPP AN and the 3GPP AAA Server. The Packet Data Network Gateway (PDN GW) provides the UE with connectivity to the external packet data networks by being the point of exit and entry of traffic for the UE. The Serving Gateway (SGW) that is located in the Visitor Public Land Mobile Network (VPLMN), routes and forwards user data packets. Evolved Packet Data Gateway (ePDG) is a gateway with which an IPsec tunnel is established by

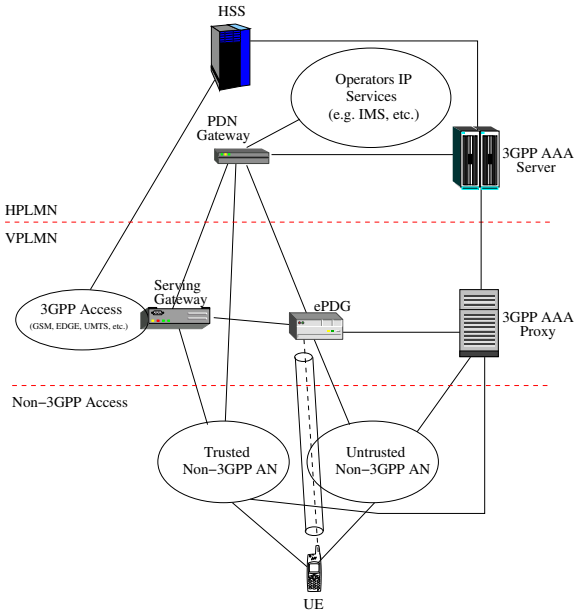


Figure 1: Security architecture for Non-3GPP access to the EPS

the UE for untrusted Non-3GPP access to EPS.

3 Access Security in Non-3GPP Access to the EPS

The AKA protocol adopted to provide access security for trusted/untrusted Non-3GPP access to EPS is Extensible Authentication Protocol for Authentication and Key Agreement (EAP-AKA) [4]. The EAP server for EAP-AKA is the 3GPP AAA Server residing in the EPC. The following subsections provides an overview of the use of EAP-AKA for trusted and untrusted Non-3GPP access.

3.1 Trusted Non-3GPP Access

In trusted Non-3GPP access, the UE connects with the EPC directly through the Non-3GPP AN. For access security, the UE and the 3GPP AAA Server executes EAP-AKA protocol between them. At the end of a successful EAP-AKA, necessary key materials for secured data communication between the UE and the Non-3GPP AN is established.

At first a connection is established between the UE and the Non-3GPP AN, using a Non-3GPP AN technology specific procedure. In order to begin the EAP-AKA procedure, the Non-3GPP AN sends an EAP Request/Identity message to the UE. In response, the UE sends an EAP Response/Identity message back to the Non-3GPP AN that contains the identity of the UE in Network Access Identifier (NAI) format [2]. The transmitted identity may either be a temporary identity allocated to the UE in the previous authentication or, in case of the first authentication, the $IMSI$. The mes-

sage is then routed towards the proper 3GPP-AAA Server through one or more AAA proxies with the help of the realm part of the NAI.

In case the NAI received from the UE contains a temporary identity, the 3GPP AAA Server extracts the corresponding $IMSI$ from it. By producing this $IMSI$, authentication data needed for mutual authentication between the UE and the 3GPP-AAA Server is acquired by the 3GPP-AAA Server from the HSS. The authentication data comprises of an Authentication Vector (AV), which is based on the authentication vectors used in UMTS [3]. It contains a random part $RAND$, an authenticator token $AUTN$ used for authenticating the network to the UE, an expected response part $XRES$, a 128-bit Integrity Key IK, and a 128-bit Cipher Key CK.

$$AV = (RAND, AUTN, XRES, IK, CK) \quad (1)$$

The $AUTN$ contains a sequence number SQN used to indicate freshness of the AV.

After an AV is acquired, the 3GPP-AAA Server derives new keying material viz. Master Session Key (MSK) and Extended Master Session Key ($EMSK$), from IK and CK. Fresh temporary identities may also be generated at this stage. The temporary identities are then encrypted and integrity protected with the keying material. The 3GPP-AAA server sends $RAND$, $AUTN$, a Message Authentication Code (MAC) (generated using the keying material) and the encrypted temporary identities to the Non-3GPP AN in an EAP Request/AKA-Challenge message. $RAND$, $AUTN$, MAC and the encrypted identities are then forwarded to the UE by the Non-3GPP AN.

The UE runs UMTS algorithm [3] on the Subscribers Identity Module (SIM). The SIM verifies that $AUTN$ is correct and hereby authenticates the network. If $AUTN$ is incorrect, the UE rejects the authentication. If $AUTN$ is correct, the UE computes RES, IK and CK. It then derives the keying material MSK and $EMSK$ from the newly computed IK and CK, and checks the received MAC with this keying material. If encrypted temporary identities were received, then the UE stores them for future authentications. The UE calculates a new MAC value covering the EAP message with the new key material. The UE then sends EAP Response/AKA-Challenge containing the calculated RES and the newly calculated MAC value to the Non-3GPP AN. The Non-3GPP AN in turn forwards the EAP Response/AKA-Challenge packet to 3GPP-AAA Server.

The 3GPP-AAA Server checks the received MAC and compares $XRES$ (received earlier from the HSS as part of AV) to the received RES. If all checks are successful, the 3GPP-AAA Server sends an EAP success message to Non-3GPP AN through a trusted link. The keying material MSK is also included in this message for Non-3GPP AN technology specific confidentiality and/or integrity protection; the Non-3GPP AN stores this keying material to be used in communication with the authenticated UE. The Non-3GPP AN informs the UE about the successful authentication with the EAP success message. This completes the EAP-AKA procedure that is required to register the UE with the Non-3GPP AN, at the end of which the UE and the authenticator in the Non-3GPP AN share keying material derived during the exchange.

3.2 Untrusted Non-3GPP Access

Unlike trusted Non-3GPP access, in untrusted Non-3GPP access, the UE connects with the EPC via the ePDG. The UE executes EAP-AKA using Internet Key Exchange version 2 (IKEv2) protocol [17] to establish an IPsec tunnel with the ePDG. The UE and the ePDG exchange a pair of messages to establish an IKEv2 channel in which the ePDG and UE negotiate cryptographic algorithms, exchange nonces and perform a Diffie Hellman exchange. In the remaining part of the authentication process, EAP-AKA (as explained in Section 3.1) is executed through this channel. After completion of the tunnel establishment and the EAP-AKA process, the UE and the ePDG share keying material that was derived during the process. The keying material is used for secured user data exchange through the tunnel during further communication between the UE and the ePDG.

3.3 Identity Privacy

In order to ensure identity privacy to the subscribers, the 3GPP-AAA Server generates and allocates temporary identities to the UE in a secured way (as discussed in Section 3.1). For identity presentation, the allocated temporary identities are transmitted by the UE instead of the permanent identity [4]. The UE does not interpret the temporary identities, it just stores them and uses them at the next authentication.

In-spite of the security measures, EAP-AKA has vulnerabilities due to which the intermediary network elements has to be entrusted with the $IMSI$ through the radio link.

3.3.1 Vulnerabilities during Trusted Non-3GPP Access

- The $IMSI$ has to be transmitted in clear text through the radio link for identity presentation during the very first authentication.
- If the 3GPP-AAA Server does not recognise a temporary identity, it will request the UE to send its permanent identity.
- A corrupt Non-3GPP AN may utilise the received $IMSI$ for various kind of malicious activities or may pass this identity to an unreliable party.
- A malicious/fake Non-3GPP AN may also take advantage of the above situation by creating a spurious EAP Request/Identity message to request the UE for its $IMSI$.

.

3.3.2 Vulnerabilities during Untrusted Non-3GPP Access

As a tunnel is established between the UE and the ePDG for secured communication during untrusted Non-3GPP access, there is no threats against identity privacy from passive attackers. However, there exist the following threats from active attackers:

- The protected channel is encrypted but not authenticated at the time of receiving the ($IMSI$). The IKEv2 messages, when using EAP, are authenticated at the end of the EAP exchange. So in case of a man-in-the middle attack, the attacker may pose as a genuine ePDG and may request the UE for the $IMSI$. Although the attack would eventually fail at the time of the authentication, the attacker would have managed to see the $IMSI$ in clear text by then.
- The $IMSI$ would be visible to the ePDG, which in roaming situations may be in the VPLMN.

4 Related Work

In mobile networks, the need to protect the identity privacy of a subscriber even from intermediary network elements like the visitor access network is well established. Herzberg *et al.* [15] pointed out that in an ideal situation no entity other than the subscriber himself and a responsible authority in the subscriber's home domain should know the real identity of the user. Even the authority in the visited network should not have any idea about the real identity.

Off late, several schemes were proposed by various researchers [7, 13, 14, 16, 19, 23]. However, none of these schemes are in line with EAP-AKA. For a mobile operator that already has a big subscriber base, changing over to a completely new authentication and key agreement protocol is a big challenge. Therefore, an ideal scheme would be the one that can be easily configured into EAP-AKA.

5 Our Proposed Security Extension

In this section, we propose a security extension where Knowledge of the $IMSI$ of a subscriber is restricted to the UE and the HSS. We propose to replace the transmission of the $IMSI$ with a Dynamic Mobile Subscriber Identity ($DMSI$). A fresh $DMSI$ is created as and when its need arises, and its value is derived from the most recent $RAND$ (Equation (1)) received during a successful EAP-AKA procedure. As a result, transmission of a $DMSI$ does not compromise the permanent identity of the user. The extension is implemented only at the Subscriber Identity Module (SIM) of the UE and the HSS. The extension can be introduced in the exiting system as an optional service, with the subscriber requiring to collect a new SIM in place of his/her existing SIM, or can be introduced on a rolling basis as new SIMs are issued. This work is based on the authors' earlier work that was proposed for UMTS [8] and LTE [10].

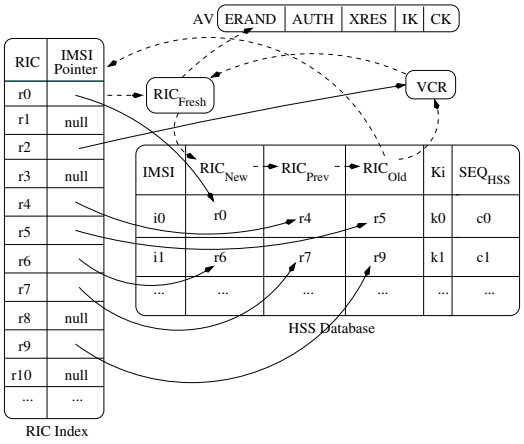


Figure 2: HSS database and the RIC index

In order to enable the UE to create a $DMSI$, a fresh random number called Random number for Identity Confidentiality (RIC) is used by the HSS. The HSS maintains a pool of $RICs$, some of which are in-use at a point of time (i.e., already assigned to different UEs). During each run of the EAP-AKA protocol, a not-in-use RIC selected from the pool is securely transferred to the UE. The selected RIC has to be sufficiently random, such that there is no correlation with a previously selected RIC. A mapping between the selected RIC and the $IMSI$ of the UE is maintained for a certain period of time (explained later in this section) at the HSS, so that the HSS can uniquely identify the UE with this RIC at a later instant. A $DMSI$ is assembled at the UE with the most recently received RIC. Thus, $DMSI$ is a function of RIC, details of which is explained later in this section (Equation (6)).

$$DMSI = f_i(RIC)$$

A RIC of size b $bits$, provide a pool of n unique RIC values. Where,

$$n = 2^b \tag{2}$$

We propose the size of RIC to be 32 bit, which is same as the size of Temporary Mobile Subscriber Identities ($TMSIs$) used in UMTS. A 32 bit RIC provides a pool of $2^{32} = 4.29$ billion (approx) unique RIC values. However, size of the RIC may even be determined by the operator depending on the anticipated subscriber base of the HSS, provided it is lesser than 128 $bits$.

In order to quickly locate a RIC in the HSS's database, a database index called *RIC-index* is maintained at the HSS (Figure 2). The *RIC-Index* contains all the $n = 2^b$ possible $RICs$ sorted according to their values. Each entry in the *RIC-Index* contains a pointer against it, which is called an *IMSI-Pointer*. This pointer either points to an $IMSI$ in the HSS's database or is *null*, depending on whether that particular RIC is allocated to an UE or is

unallocated at a particular instance of time. The collection of all the $RICs$ in the *RIC-Index* having a *null* value against it, forms a pool of not-in-use $RICs$, whereas the rest of the $RICs$ in the *RIC-Index* that points to some $IMSI$, represents the $RICs$ that are in-use. The total number of entries in the *RIC-Index* is fixed at n, irrespective of the number of $RICs$ that are currently in-use in the HSS's database. Even though such an index would require more disk space (Section 8), compared to an index whose size grows and shrinks according to the number of $RICs$ that are in-use at a particular instance in the HSS's database, it relieves the HSS of computational overhead involved during frequent insertions and deletions in the index.

In order to allocate a fresh RIC to the UE during every run of the EAP-AKA, a RIC called RIC_{Fresh} is chosen randomly from the pool of not-in-use $RICs$ at the HSS. RIC_{Fresh} is then cryptographically embedded into the $RAND$ part of the AV (Equation (1)) that reaches the UE as a challenge in due course of the AKA mechanism (as explained in Section 3). This resultant random number after embedding RIC into $RAND$ taking Ki as parameter is referred to as Embedded RAND ($ERAND$).

$$ERAND = f_{Embed_{Ki}}(RIC_{Fresh}, RAND) \quad (3)$$

The modified AV after embedding RIC into $RAND$ looks like the following:

$$AV = (ERAND, AUTH, XRES, IK, CK) \quad (4)$$

This $ERAND$ is now used by the 3GPP-AAA server, instead of the $RAND$ (as in the original EAP-AKA), to challenge the UE. Since the size of $RAND$ and $ERAND$ is same (i.e., 128 *bit*), the 3GPP-AAA server will not be able to perceive this change and will continue as before. Example algorithms to embed a 32 *bit* RIC into a 128 *bit* $RAND$ and to extract the embedded RIC form the $ERAND$ is proposed in [9]. Only the UE having knowledge of the long term shared key Ki is capable of extracting RIC from $ERAND$.

$$RIC = f_{Extract_{Ki}}(ERAND)$$

Multiple copies (m) of $RICs$ - the fresh and few previously generated $RICs$, stored in the fields: $RIC_{New}, RIC_{Prev}, RIC_{Old}$, etc. - are maintained at the HSS's database against a particular $IMSI$ along with the long term secret key Ki. This ensures robustness of the protocol even when an $ERAND$ gets lost in transition and does not reach the UE (Figure 2). Such an arrangement ensures that a mapping between the RIC that is currently stored at the UE and the corresponding $IMSI$ is always maintained at the HSS. However, like any other critical information such as the subscriber's security credentials, billing details, etc., in case of the $RICs$ maintained in the HSS's database also, it is the responsibility of the operator to have a robust backup mechanism against database crash.

We propose the value of m to be 4, i.e., $m = 4$; however, an operator may choose to have their own value for m. In UMTS, the size of the Temporary Mobile Subscriber Identity ($TMSI$) is 32 *bit*, out of which 2 *bits* are used to identify the type of identity (i.e., circuit switched, packet switched, etc.). The remaining 30 *bits* with 2^{30} possibilities are considered sufficient to allocate temporary identities to all the subscribers within a network. Therefore, in our proposal, $2^2 = 4$ is selected as the value for m. This will enable the HSS to have at the most $2^{30} = 1.073$ billion (approx) subscribers, which is 5.73 times more than the 187.302 million (approx) subscriber base of the largest cellular operator in India as of June, 2012 [20]). Thus, if s is the maximum number of subscribers that the operator wants the proposed extension to handle, then

$$s = n/m \quad (5)$$

where, n is the total number of possible $RICs$ in the entire pool and m is the number of $RICs$ maintained against each $IMSI$ in the HSS's database.

In order to verify the freshness of a received $DMSI$ and to prevent replay attacks, the HSS maintains a field called SEQ_{HSS} against every $IMSI$ in its database. SEQ_{HSS} is used to store the sequence number of the most recent $DMSI$ received from the UE.

5.1 Resolving a DMSI to an IMSI

In EAP-AKA, during situations (as explained in Sections 3.3.1 and 3.3.2) where the $IMSI$ needs to be transmitted by the UE, we propose to transmit a $DMSI$ (in NAI format) instead of the $IMSI$. The $DMSI$ is created using the RIC extracted from the most recent $ERAND$ received by the UE during an EAP-AKA as follows:

$$DMSI = MCC \| MNC \| RIC \| ERIC \quad (6)$$

where, MCC stands for the Mobile Country Code, MNC stands for the Mobile Network Code, and $ERIC$ is created by encrypting a padded RIC (say RIC_{padded}) with the Advanced Encryption Standard (AES) algorithm, taking the long term secret key Ki as parameter. Thus,

$$ERIC = f_{n_{Ki}}(RIC_{padded})$$

where,

$$RIC_{padded} = RIC \| SEQ_{UE} \| R$$

SEQ_{UE} is the value of a 32 bit counter that is maintained at the UE; whenever a new $DMSI$ is created for identity presentation, $SEQ_{UE}{}'s$ value is incremented by one. R is a $128 - (32 + b)$ bit random number. The inclusion of SEQ_{UE} ensures freshness of the $DMSIs$, whereas the inclusion of R completes the block size of 128 bits that is necessary to be fed into the AES cipher. In addition, R introduces sufficient amount of randomness to harden cryptanalysis of the ciphertext. The realm part of the $DMSI$ in NAI format helps the intermediate AAA proxy servers to guide the request to the appropriate 3GPP-AAA server. The 3GPP-AAA server treats the received

$DMSI$ as an $IMSI$ and therefore forwards the $DMSI$ along with a request for AV to the HSS. Thus, the onus of resolving the $DMSI$ is passed on to the HSS. On receipt of the request for AV, the HSS executes a sequence of instructions. First and foremost, the HSS resolves the $DMSI$, which is done by locating the RIC part of the received $DMSI$ in the *RIC-Index* and by mapping it to the corresponding $IMSI$ through the *IMSI-Pointer*. The $ERIC$ part of the $DMSI$ is then decrypted using AES and the corresponding key Ki. Thus,

$$RIC_{padded} = f_{dKi}(ERIC)$$

The RIC contained in RIC_{padded} is compared with the RIC part of the $DMSI$, the success of this comparison ensures that a malicious agent did not create the $DMSI$. The SEQ_{UE} part of RIC_{padded} is then compared with the value stored against SEQ_{HSS} field in the HSS's database. If $SEQ_{UE} > SEQ_{HSS}$, the request is proven as a fresh request. Failure of any of these two comparisons, leads to rejection of the request. If the request for AV is found to be fresh and from a genuine source (from the above comparisons), SEQ_{UE} is copied into SEQ_{HSS}

$$SEQ_{HSS} = SEQ_{UE}$$

and a fresh AV (Equation (1)) is generated using the procedure used in EAP-AKA.

5.2 Embedding a RIC into the RAND Part of AV

Whenever a RIC needs to be embedded into a $RAND$ at the HSS, a new RIC (RIC_{Fresh}) is selected from the pool of not-in-use $RICs$. In order to select RIC_{Fresh}, a b *bit* random number (say RN) is generated using a standard Pseudo Random Number Generator (PRNG). For this, we propose to use National Institute of Standards and Technology (NIST) recommended random number generator based on ANSI X9.31 Appendix A.2.4 Using AES [18], which appears in the list of approved random number generators for Federal Information Processing Standards Publication (FIPS PUB) 140-2 [11]. With a 128 *bit* key, this PRNG generates a 128 *bit* random number, the b most significant *bits* of which is selected as RN.

$$RN = f_{PRNG}(seed)$$

This RN is then searched for in the *RIC-Index*. If the *IMSI-Pointer* against RN in the *RIC-Index* is found to be *null*, RN is selected as RIC_{Fresh} and the *null* value is replaced with the address of the record in the HSS's database where the $IMSI$ is stored.

$$\begin{aligned} RIC_{Fresh} &= RN \\ RN.IMSI\text{-}Pointer &= Address\ of\ IMSI \end{aligned}$$

The oldest RIC value (i.e., RIC_{Old}) stored against the $IMSI$ is then returned to the pool of not-in-use RIC by searching for it in the *RIC-Index* and by setting the *IMSI-Pointer* against it to *null*.

$$RIC_{Old}.IMSI\text{-}Pointer = null$$

In case the *IMSI-Pointer* against RN in the *RIC-Index* is not *null*, it may be inferred that there is a collision, and RN is currently in-use. For collision resolution, a b *bit* variable called Variable for Collision Resolution (VCR) is used (Figure 2). The VCR contains a not-in-use RIC; an indication of this fact is specified in the *RIC-Index* by setting the *IMSI-Pointer* against the value in VCR to the address of VCR. At the very outset, during initialisation of the HSS's database, a b *bit* random number (say RN_0) is stored in the VCR and the *IMSI-Pointer* against it in the *RIC-Index* is set to the address of VCR.

$$\begin{aligned} RN_0 &= f_{PRNG}(seed) \\ VCR &= RN_0 \\ RN_0.IMSI\text{-}Pointer &= Address\ of VCR. \end{aligned}$$

Whenever there is a collision, the b *bit* value stored in the VCR is selected as RIC_{Fresh}. VCR is then searched for in the *RIC-Index* and the *IMSI-pointer* against it in the *RIC-Index* is made to point to the record in the HSS's database where the $IMSI$ is stored.

$$\begin{aligned} RIC_{Fresh} &= VCR \\ VCR.IMSI\text{-}Pointer &= Address\ of\ IMSI \end{aligned}$$

In order to replace the RIC stored in the VCR with a fresh RIC, the oldest RIC (i.e., RIC_{Old}) stored against the $IMSI$ is copied into VCR. RIC_{Old} is then searched for in the *RIC-Index* and the *IMSI-pointer* against it is set to the address of VCR.

$$\begin{aligned} VCR &= RIC_{Old} \\ RIC_{Old}.IMSI\text{-}Pointer &= Address\ of\ VCR \end{aligned}$$

The above procedure used to refresh the VCR introduces ample entropy to make the selection procedure of RIC even more random, because it is impossible to predict which $IMSI's\ RIC_{Old}$ value will refresh the VCR during the next EAP-AKA at the HSS. It solely depends on the call timing and usage pattern of all the active subscribers registered with the HSS. Moreover, the distribution process of the $RICs$ itself is random.

RIC_{Fresh} is then embedded into the $RAND$ part of AV (using Equation (3)). A copy of RIC_{Fresh} is also stored against the $IMSI$ in the HSS's database. To make space for RIC_{Fresh}, the value in RIC_{Prev} is copied into RIC_{Old} and the value in RIC_{New} is copied into RIC_{Prev}. Finally, the value in RIC_{Fresh} is copied into RIC_{New}.

$$\begin{aligned} RIC_{Old} &= RIC_{Prev} \\ RIC_{Prev} &= RIC_{New} \\ RIC_{New} &= RIC_{Fresh}. \end{aligned}$$

The AV is then send to the 3GPP-AAA server. The 3GPP-AAA server in turn, forwards a challenge containing $ERAND$ and $AUTH$ (extracted from AV) to the UE

Table 1: Functions used in the extension

Function	Details
f_i	Generates a $DMSI$ from a given RIC.
f_{Embed}	Embeds a 32 bit RIC into a 128 bit $RAND$.
$f_{Extract}$	Extracts the 32 bit RIC from a 128 bit $ERAND$.
f_n	Encrypts RIC_{Padded} to find $ERIC$.
f_d	Decrypts $ERIC$ to find RIC_{Padded}.
f_{PRNG}	Generates a 128 bit pseudo random number.

(through the Non-3GPP AN). The rest of the process continues in the same way as EAP-AKA. After successful authentication, the UE stores the $ERAND$ received as a challenge in its flash memory, to be used for identity presentation during subsequent authentications. The RIC embedded in $ERAND$ is extracted only when a $DMSI$ needs to be assembled.

An $ERAND$ (Say $ERAND_{First}$) that has a unique RIC called RIC_{First} embedded into it, is stored in the SIM's flash memory before a subscriber procures it from the service provider. RIC_{First} is also stored at the HSS's database and an entry in the *RIC-Index* is made accordingly. RIC_{First} is meant for one time usage for $DMSI$ creation during the very first authentication in the SIM's life time.

In some exceptional situations like failure of an ongoing EAP-AKA or due to an active attack by an adversary, the UE may not receive the next RIC (from the HSS) after it has already used the most recently received RIC to create and transmit a $DMSI$. In such a situation, if the need to transmit a $DMSI$ arises again, the UE can reuse the most recently received RIC to create the next $DMSI$. This can continue, as long as the UE does not receive a fresh RIC from the HSS (during a successful EAP-AKA). Even though such a recovery mechanism, in the worst case, may allow an adversary to link two or more failed EAP-AKA of the same UE, an adversary cannot gain anything from this, in terms of compromised identity privacy. Moreover, it is a much better option than transmitting the $IMSI$ itself.

A summary of all the functions used in the security extension is presented in Table 1.

5.3 Achievements

The key achievements of the proposed extension may be summarised as follows:

- *End to end user identity privacy*: Knowledge of $IMSI$ is confined only to the UE and the HSS.
- *Relaxed trust requirement*: Since the $IMSI$ is never revealed to the Non-3GPP AN or the ePDG, the HSS to Non-3GPP AN and HSS to ePDG trust relationship requirement with respect to $IMSI$ is relaxed. This relaxation will facilitate interoperability.
- *No overhead at the intermediary network*: The proposed security extension has to be implemented only at the UE and the HSS, intermediary elements like the Non-3GPP AN and the AAA servers can continue to maintain status quo. Thus, for an operator that adapts this extension, there is no additional cost of negotiation, implementation, computation, etc., to get the intermediary agents (that may even belong to third party operators) on board.
- *Can be adopted as an extension*: The proposed extension is in line with EAP-AKA and can be adopted as an extension.

Since, with the proposed extension, the $IMSI$ is never transmitted at any stage of the EAP-AKA protocol, all the vulnerabilities listed in Section 3.3.1 and Section 3.3.2 are eliminated; thereby relaxing the need to trust an intermediary network element with the permanent identity of a subscriber. Thus, with respect to signalling data (that does not reveal the $IMSI$ any more), the extension removes the need to establish a tunnel between the UE and the EPC during an untrusted Non-3GPP access. However, the need of the tunnel with respect to user data continues to exist.

6 Formal Analysis

We performed a formal analysis of the proposed scheme through an enhanced BAN logic [6] called AUTLOG [21]. A similar analysis is performed by 3GPP in [1]. Through this analysis, the security goals described in the following subsection are proven to be achieved by the proposed scheme.

6.1 Security Goals

$IMSI$ should be a shared secret between the UE and the HSS. The same should not be disclosed by the UE to any third party including the Non-3GPP AN.

G1: $UE \text{ believes } UE \overset{IMSI}{\longleftrightarrow} HSS$

When ever temporary identities fails to protect the permanent identity, a backup mechanism is followed according to our proposed extension, so that identity privacy may still be ensured to the subscriber. According to this mechanism (Section 5), a $DMSI$ (created with the RIC that is extracted from the most recent $RAND$ received by the UE) is transmitted in lieu of the $IMSI$. During every successful run of the EAP-AKA protocol, if the UE receives a fresh RIC, it can easily protect its permanent identity.

G2: $UE \text{ believes } UE \text{ has } RIC$

G3: $UE\ believes\ fresh(RIC)$

It should not be possible for anyone except the HSS (that has access to the RIC-Index) to map a $DMSI$ with its corresponding $IMSI$.

G4: $UE\ believes\ \neg(DMSI \equiv IMSI)$

6.2 Prerequisites

The UE recognises Ki and believes that it is a good key for communication with the HSS:

$$UE\ has\ Ki \tag{7}$$

$$UE\ recognises\ Ki \tag{8}$$

$$UE\ believes\ HSS \overset{Ki}{\longleftrightarrow} UE \tag{9}$$

Since the UE is capable of verifying freshness of SEQ contained in the $AUTN$ part of the received challenge, it believes in $SEQ's$ freshness.

$$UE\ believes\ fresh(SEQ) \tag{10}$$

The UE regards $ERAND$ as an atomic message.

$$(ERAND)_{UE} \equiv ERAND \tag{11}$$

The challenge received by the UE contains a Message Authentication Code (MAC) in the $AUTN$ part of the challenge. MAC is an encryption of SEQ and $ERAND$ with key Ki. The UE believes that it has not said MAC itself.

$$UE\ believes\ \neg(UE\ said\ enc(Ki, SEQ, ERAND)) \tag{12}$$

The UE believes that the HSS controls the freshness of RIC and that if the HSS says $ERAND$ along with an $AUTN$ with a fresh SEQ in it, the RIC contained in the $ERAND$ is also fresh.

$$UE\ believes\ HSS\ controls\ fresh(RIC) \tag{13}$$

$$UE\ believes\ (HSS\ says\ (SEQ, ERAND) \longrightarrow HSS\ believes\ fresh(RIC)) \tag{14}$$

$ERAND$ is an encrypted form of RIC. With knowledge of Ki, the UE can easily extract RIC from $ERAND$. Thus, UE is able to identify $ERAND$ with $enc(Ki, RIC)$.

$$(ERAND)_{UE} \equiv enc(Ki, RIC) \tag{15}$$

UE believes that HSS has jurisdiction and belief concerning the $IMSI$ as a shared secret between the UE and the HSS.

$$UE\ believes\ HSS\ controls\ HSS \overset{IMSI}{\longleftrightarrow} UE \tag{16}$$

$$UE\ believes\ HSS\ believes\ HSS \overset{IMSI}{\longleftrightarrow} UE \tag{17}$$

UE believes that HSS has jurisdiction on the fact that without access to the *RIC-Index*, RIC cannot be linked in any way with the corresponding $IMSI/MSIN$:

$$UE\ believes\ HSS\ controls\ \neg(RIC \equiv IMSI) \tag{18}$$

$$UE\ believes\ (HSS\ says\ ERAND \longrightarrow HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN)) \tag{19}$$

$$UE\ believes\ (HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN) \longrightarrow HSS\ believes\ \neg(RIC \equiv MSIN)) \tag{20}$$

UE sees the following:

$$UE\ sees\ ERAND, \{SEQ\}_{enc(Ki,ERAND)}, enc(Ki, SEQ, ERAND) \tag{21}$$

6.3 Proving the Security Goals

Figure 3 depicts step by step deduction of the security goals. Each circle in the figure represents an equation in this section. The label against each circle represents the inference rule/rules of AUTLOG that is/are used to derive the particular equation. The following $E(\#)$ denotes the Equation $(\#)$.

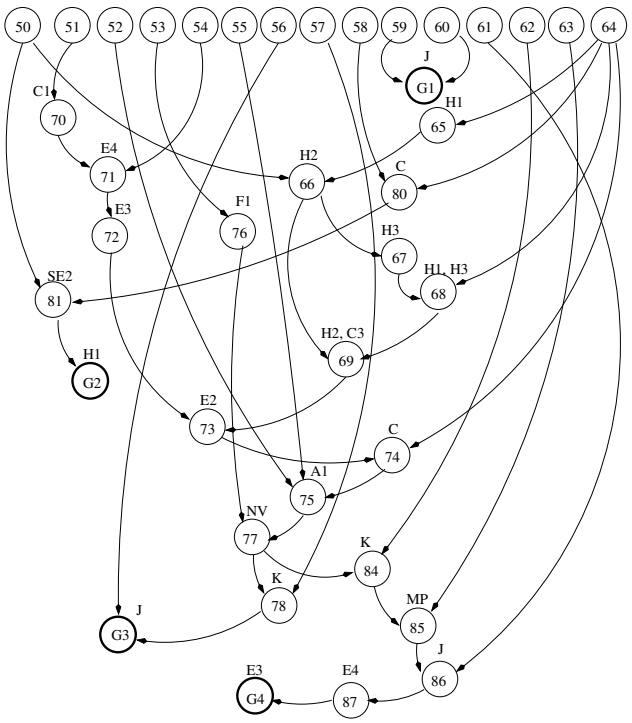


Figure 3: Deduction of security goals

$$E(21) \overset{H1}{\longrightarrow} UE\ has\ ERAND \tag{22}$$

$$E(7), E(22) \overset{H2}{\longrightarrow} UE\ has\ (Ki, ERAND) \tag{23}$$

$$E(23) \xrightarrow{H3} UE\ has\ enc(Ki, ERAND) \quad (24)$$

$$E(21), E(24) \xrightarrow{H1,H3} UE\ has\ SEQ \quad (25)$$

$$E(23), E(25) \xrightarrow{H2,C3} (enc(Ki, SEQ, ERAND))_{UE} \equiv enc((Ki, SEQ, ERAND)_{UE}) \quad (26)$$

$$E(8) \xrightarrow{C1} (Ki, SEQ, ERAND)_{UE} \quad (27)$$
$$\equiv ((Ki)_{UE}, (SEQ)_{UE}, (ERAND)_{UE})$$

$$E(11), E(27) \xrightarrow{E4} (Ki, SEQ, ERAND)_{UE} \quad (28)$$
$$\equiv (Ki, SEQ, ERAND)$$

$$E(28) \xrightarrow{E3} enc((Ki, SEQ, ERAND)_{UE}) \quad (29)$$
$$\equiv enc(Ki, SEQ, ERAND)$$

$$E(26), E(29) \xrightarrow{E2} (enc(Ki, SEQ, ERAND))_{UE}$$
$$\equiv enc(Ki, SEQ, ERAND) \quad (30)$$

$$E(21), E(30) \xrightarrow{C} UE\ believes\ UE\ sees\ enc(Ki, SEQ, ERAND) \quad (31)$$

$$E(31), E(9), E(12) \xrightarrow{A1} UE\ believes\ HSS\ said\ (SEQ, ERAND) \quad (32)$$

$$E(10) \xrightarrow{F1} UE\ believes\ fresh(SEQ, ERAND) \quad (33)$$

$$E(32), E(33) \xrightarrow{NV} UE\ believes\ HSS\ says\ (SEQ, ERAND) \quad (34)$$

$$E(34), E(14) \xrightarrow{K} UE\ believes\ HSS\ believes\ fresh(RIC) \quad (35)$$

$$E(13), E(35) \xrightarrow{J} \boxed{UE\ believes\ fresh(RIC)}\ (\mathbf{G3}) \quad (36)$$

$$E(21), E(15) \xrightarrow{C} UE\ believes\ UE\ sees\ enc(Ki, RIC) \quad (37)$$

$$E(37), E(7) \xrightarrow{SE2} UE\ believes\ UE\ sees\ RIC \quad (38)$$

$$E(38) \xrightarrow{H1} \boxed{UE\ believes\ UE\ has\ RIC}\ (\mathbf{G2}) \quad (39)$$

$$E(16), E(17) \xrightarrow{J} \boxed{UE\ believes\ (HSS \xleftrightarrow{IMSI} UE)}\ (\mathbf{G1}) \quad (40)$$

$$E(34), E(19) \xrightarrow{K} UE\ believes\ HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN) \quad (41)$$

$$E(41), E(20) \xrightarrow{MP} UE\ believes\ HSS\ believes\ \neg(RIC \equiv MSIN) \quad (42)$$

$$E(18), E(42) \xrightarrow{J} UE\ believes\ \neg(RIC \equiv MSIN) \quad (43)$$

$$E(43) \xrightarrow{E4} UE\ believes\ \neg(MCC, MNC, RIC \equiv MCC, MNC, MSIN) \quad (44)$$

$$E(44) \xrightarrow{E3} \boxed{UE\ believes\ \neg(DMSI \equiv IMSI)}\ (\mathbf{G4}) \quad (45)$$

Hence, it is proven that the proposed extension meets its security goals.

7 Computational Overhead

In this section, we analyse the computational overhead of the proposed extension, using a methodology proposed in [12]. The core idea of this methodology is to determine the amount of basic operations required for implementation of an algorithm, reducing all other operations in terms of these basic operations. For overhead analysis of the proposed extension, all the other operations used in the extension are reduced to the following basic operations: Byte-wise AND, byte-wise OR, shift (bytes) and logical comparison operation. For XOR operations, we exploit the rule that a simple bit-wise XOR of x and y is equal to $x'y + y'x$. Since negations are negligible compared to AND/OR logical operations, a bit-wise XOR is considered as the sum of two bit-wise ANDs and one bit-wise OR. The methodology can be used to calculate the computational overhead of some of the key computations involved in the extension as follows:

1) *Encryption/Decryption with AES*: Let $N_{EncryptAES}$ and $N_{DecryptAES}$ be the number of basic operations needed by AES to encrypt a 128 *bit* plain-text and to decrypt a 128 *bit* cipher-text, respectively, using a 128 *bit* key. Granelli et. al. [12] found that 1720 byte-wise AND, 1268 byte-wise OR and 408 shift (byes) are involved in a 128/128 AES encryption, whereas, 5176 byte-wise AND, 3860 byte-wise OR and 1272 shift (byes) are involved in a 128/128 AES decryption. Thus,

$$N_{EncryptAES} = 3396$$
$$N_{DecryptAES} = 10308$$

2) *Encrypt RIC*: Let $N_{EncryptRIC}$ be the number of basic operations needed to encrypt a RIC to form an $ERIC$. 128/128 AES algorithm is used to carry out this encryption. Therefore,

$$N_{EncryptRIC} = N_{EncryptAES}$$

3) *Decrypt ERIC*: Let $N_{DecryptERIC}$ be the number of basic operations needed to decrypt an $ERIC$ to find a RIC. 128/128 AES algorithm is used to carry out this decryption. Therefore,

$$N_{DecryptERIC} = N_{DecryptAES}$$

4) *Search RIC*: Let $N_{SearchRIC}$ be the number of basic operations needed to search a *RIC* in the *RIC-Index*. The *RIC-Index* contains $n = 2^b$ number of entries arranged in sequential order, where b is the size of the *RIC* in bits. A *RIC* can be searched using binary search in $O(log_2 n)$ logical comparison operations. Thus, from Equation (5)

$$\begin{aligned} N_{SearchRIC} &= O(log_2 n) \\ &= O(log_2(m \times s)) \end{aligned}$$

5) *Select* RIC_{Fresh}: Let $N_{SelectRIC_{Fresh}}$ be the number of basic operations needed to select a not-in-use *RIC* as RIC_{Fresh}. At first a b *bit* random number (RN) is generated, using a standard PRNG (in say N_{Rn} number of operations). For this purpose, the PRNG based on ANSI X9.31 Using AES can be used. In this PRNG, 256 bit-wise XOR operations (which amounts to 64 byte-wise AND and 32 byte-wise OR operations) and 3 rounds of the AES encryption algorithm are performed to generate a pseudo random number [18]. Thus,

$$\begin{aligned} N_{Rn} &= 64 + 32 + 3 \times N_{EncryptAES} \\ &= 10284 \end{aligned}$$

RN is then searched in *RIC-Index* in $N_{SearchRIC}$ number of operations. If the pointer against RN is *null* (with this comparison requiring 1 comparison operation), then RN is selected as RIC_{Fresh} by setting the *IMSI-Pointer* against it to the address of the concerned $IMSI$. Otherwise, the value in VCR is selected as RIC_{Fresh}. VCR is then searched in the *RIC-index* in $N_{SearchRIC}$ number of operations and the *IMSI-Pointer* against it is set to the address of the concerned $IMSI$. Thus,

$$\begin{aligned} N_{SelectRIC_{Fresh}} &= N_{Rn} + 2 \times N_{SearchRIC} + 1 \\ &= 10284 + 2 \times O(log_2(m \times s)) + 1 \\ &= 10285 + 2 \times O(log_2(m \times s)) \end{aligned}$$

6) *Embed RIC into RAND*: Let $N_{EmbedRIC}$ be the number of basic operations needed to embed a *RIC* into a $RAND$. Considering the example algorithm proposed in [9], we found that a total of 32 bit-wise XOR operations (which amounts to 8 byte-wise AND and 4 byte-wise OR operations) and 1 round of the AES encryption algorithm are performed to embed a 32 *bit RIC* into a 128 *bit* $RAND$. Thus,

$$\begin{aligned} N_{EmbedRIC} &= 12 + N_{EncryptAES} \\ &= 3408 \end{aligned}$$

7) *Extract RIC from ERAND*: Let $N_{ExtractRIC}$ be the number of basic operations needed to extract the embedded *RIC* from an $ERAND$. Considering the algorithm proposed in [9], we found that a total of 32 bit-wise XOR operations (which amounts to 8 byte-wise AND and 4 byte-wise OR operations) and 1 round of the AES decryption algorithm are performed to extract the 32 *bit RIC* from a 128 *bit* $ERAND$. Thus,

$$\begin{aligned} N_{ExtractRIC} &= 8 + 4 + N_{DecryptAES} \\ &= 10320 \end{aligned}$$

8) *Return* RIC_{Old}: RIC_{Old} is searched in the *RIC-Index* in $N_{SearchRIC}$ operations. RIC_{Old} is then returned to the pool of not-in-use $RICs$ by setting the *IMSI-Pointer* against RIC_{Old} in the *RIC-Index* to either *null* or the address of VCR, depending on whether RN was selected as RIC_{Fresh} or VCR was selected as RIC_{Fresh}. Thus, if ($N_{ReturnRIC_{Old}}$) is the total time taken for this purpose,

$$\begin{aligned} N_{ReturnRIC_{Old}} &= N_{SearchRIC} \\ &= O(log_2(m \times s)) \end{aligned}$$

7.1 Computational Overhead at the UE

The proposed extension provides a backup mechanism that is used to identify the subscriber in situations where the temporary identities fail to identify the subscriber. According to this mechanism, a $DMSI$ is transmitted to the 3GPP-AAA Server in lieu of the permanent identity (i.e., the $IMSI$). The following computations are introduced at the UE when the UE identifies itself with a $DMSI$:

1) Extract *RIC* from the most recently received $ERAND$ in time say $T_{ExtractRIC}$.
2) Create $ERIC$ from the extracted *RIC* in time say $T_{EncryptRIC}$.

Thus, the computational overhead (say N_{UE}) introduced at the UE when a $DMSI$ is transmitted, can be calculated as follows:

$$\begin{aligned} N_{UE} &= N_{ExtractRIC} + N_{EncryptRIC} \\ &= 12 + N_{DecryptAES} + N_{EncryptAES} \\ &= 13716 \end{aligned}$$

Therefore, the overall computational overhead introduced at the UE by the extension is as follows:

$$N_{UE} = \begin{cases} 0 & \text{-when a temporary identity} \\ & \text{is transmitted,} \\ 13716 & \text{-when a } DMSI \text{ is transmitted.} \end{cases} \quad (46)$$

7.2 Computational Overhead at the HSS

During every EAP-AKA, the extension requires the HSS to embed a fresh *RIC* into the $RAND$ part of the AV. In addition, when a $DMSI$ is used for identity presentation, the HSS needs to resolve the $DMSI$.

The following computations (executed in say $N_{Resolve}$ number of operations) are introduced at the HSS when a $DMSI$ has to be resolved:

1) Search RIC in the *RIC-Index* in $N_{SearchRIC}$ operations.

2) Decrypt $ERIC$ to find RIC in $N_{DecryptRIC}$ operations.

3) Compare the decrypted RIC with the RIC contained in the $DMSI$ in 1 comparison operation.

4) Compare SEQ_{UE} and SEQ_{HSS} in 1 comparison operation.

$$\begin{aligned} N_{Resolve} &= N_{SearchRIC} + N_{DecryptRIC} + 1 + 1 \\ &= O(log_2(m \times s)) + N_{DecryptAES} + 2 \\ &= O(log_2(m \times s)) + 10310 \end{aligned} \quad (47)$$

The following computations (executed in say N_{Embed} number of operations) are introduced at the HSS, when RIC_{Fresh} has to be embedded into the $RAND$ part of AV.

1) Select RIC_{Fresh} in $N_{SelectRIC_{Fresh}}$ operations.

2) Embed RIC_{Fresh} into $RAND$ in $N_{EmbedRIC}$ operations.

3) Return the RIC_{Old} to the pool of not-in-used $RICs$ and store RIC_{Fresh} in the HSS's database in $N_{ReturnRIC_{Old}}$ operations.

Thus,

$$\begin{aligned} N_{Embed} &= N_{SelectRIC_{Fresh}} + N_{EmbedRIC} \\ &\quad + N_{ReturnRIC_{Old}} \\ &= 10285 + 2 \times O(log_2(m \times s)) + \\ &\quad 3408 + O(log_2(m \times s)) \\ &= 13693 + 3 \times O(log_2(m \times s)) \end{aligned} \quad (48)$$

The total computational overhead introduced at the HSS by the extension is equal to N_{Embed}, when a temporary identity is received at the HSS; whereas, it is equal to $N_{Resolve} + N_{EmbedRIC}$, when a $DMSI$ is received at the HSS.

Since we proposed the value of b to be 32 *bit* and the value of m to be 4 (Section 3), the value of $O(log_2(m \times s))$ can be derived using Equation (5) and Equation (2) as follows:

$$\begin{aligned} O(log_2(m \times s)) &= O(log_2(n)) \\ &= O(b) \\ &= 32 \end{aligned}$$

Thus, the overall computational overhead (say N_{HSS}) introduced at the HSS is as follows:

$$N_{HSS} = \begin{cases} 13789 & \text{-when a temporary identity} \\ & \text{is received,} \\ 24131 & \text{-when a } DMSI \text{ is received.} \end{cases}$$

Table 2: Summary of overhead analysis

Overhead	UE	HSS
Computational overhead when temporary identity is transmitted/received	0	13789
Computational overhead when $DMSI$ is transmitted/received	13716	24131
Time complexity	$O(1)$	$O(log_2(s))$
Space overhead	160 *bit*	$416 \times s$ *bit*
Communication overhead	0	0

7.3 Time Complexity

In this section, we derive the time complexity (in terms of the growth in subscriber base S) that an operator can expect at the UE and the HSS when the proposed security extension is adopted. From Equation (46), we can infer that the following time complexity is introduced at the UE.

$$T_{UE} = O(1)$$

From Equation (47) and Equation (48), the time complexity introduced at the HSS can be derived as follows:

$$\begin{aligned} T_{HSS} &= O(log_2(m \times s)) \\ &= O(log_2(s)), \quad m \text{ being a constant.} \end{aligned}$$

where, s is the maximum number of subscribers that the extension is expected to handle.

8 Space Overhead

8.1 Space Overhead at the UE

In order to store the most recently received $ERAND$, the UE needs 128 *bit* of space in the UE's flash memory. And, in order to maintain a 32 *bit* $DMSI$ counter, an additional 32 *bit* are required. Thus, the amount of space (say S_{UE}) required by the extension at the UE is:

$$S_{UE} = 160 \text{ } bit$$

8.2 Space Overhead at the HSS

Against every $IMSI$ in the HSS's database, $m = 4$ $RICs$ and a 32 *bit* SEQ_{HSS} value are stored. Thus, every record in the HSS's database will need an additional 160 *bit* space. In order to have provision for the maximum number of subscribers, i.e., s, the total amount of additional space (say S_{RIC}) needed at the HSS's database is:

$$S_{RIC} = 160 \times s \text{ } bit$$

In the *RIC-Index* there are n entries. Each of these entries has a 32 *bit RIC* and a 32 *bit IMSI-pointer*, with a total of 64 *bit*. Thus, the total amount of space (say $S_{RIC\text{-}Index}$) needed at the HSS's memory to accommodate the *RIC-Index* is:

$$\begin{aligned} S_{RIC\text{-}Index} &= 64 \times m \times s \ bit \\ &= 256 \times s \ bit, \quad \text{considering } m \text{ as } 4. \end{aligned}$$

Thus, the total amount of space (say S_{Total}) required by the extension at the the HSS is:

$$\begin{aligned} S_{Total} &= S_{RIC} + S_{RIC\text{-}Index} \\ &= 416 \times s \ bit \end{aligned}$$

where, s is the maximum number of subscribers that the extension is expected to handle.

9 Communication Overhead

In order to exchange information among the agents for smooth functioning of the extension, no additional messages are introduced to the original EAP-AKA protocol. Instead, the information are embedded into the existing messages of the EAP-AKA protocol. The procedure used for embedding the information is such that the format and size of the original messages remains the same. Thus, no communication overhead is imposed by the proposed extension at any of the agents involved in the EAP-AKA protocol.

Various overheads calculated in Section 7 through Section 9 are summarised in Table 2.

10 Conclusion

A factor that complicates Non-3GPP access to the EPS is the trust requirement on intermediary network elements like Non-3GPP AN and ePDG with respect to the subscriber's identity privacy. In this paper, we have put forward a security extension that improves the situation by taking an end to end approach, where the Non-3GPP AN or for that matter any intermediary element, need not be trusted with the permanent identity of the subscriber. This trust relaxation will not only facilitate interoperability, but also would enhance identity privacy of the subscriber. The strength of the extension lies in the fact that it can be adopted as an extension to the existing security mechanism. Moreover, it has to be implemented only at an operators level without tasking the intermediary network elements. Results of formal analysis and computational cost analysis show that the extension meets its security goals and is feasible with the existing infrastructure.

References

[1] 3GPP, "Formal Analysis of the 3G Authentication Protocol," TR 33.902, 3rd Generation Partnership Project (3GPP), 2001.

[2] 3GPP, "Numbering, addressing and identification," TS 23.003, 3rd Generation Partnership Project (3GPP), 2011.

[3] 3GPP, "3G Security; Security architecture," TS 33.102, 3rd Generation Partnership Project (3GPP), 2012.

[4] 3GPP, "3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses," TS 33.402, 3rd Generation Partnership Project (3GPP), 2012.

[5] 3GPP, "Architecture enhancements for non-3GPP accesses," TS 23.402, 3rd Generation Partnership Project (3GPP), 2012.

[6] 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, no. 1871, pp. 233–271, 1989.

[7] C. Chen, D. He, S. Chan, J. Bu, Y. Gao and R. Fan, "Lightweight and provably secure user authentication with anonymity for the global mobility network," *International Journal of Communication Systems*, vol. 24, no. 3, pp. 347–362, 2011.

[8] H. Choudhury, B. Roychoudhury and D. K. Saikia, "End-to-end user identity confidentiality for umts networks," in *2010 3rd IEEE International Conference on Computer Science and Information Technology*, vol. 2, pp. 46–50, 2010.

[9] H. Choudhury, B. Roychoudhury and D. K. Saikia, "Umts user identity confidentiality: An end-to-end solution," in *2011 Eighth International Conference on Wireless and Optical Communications Networks*, pp. 1–6, 2011.

[10] H. Choudhury, B. Roychoudhury and D. K. Saikia, "Enhancing user identity privacy in LTE," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 949–957, 2012.

[11] R. J. Easter and F. Carolyn, "Annex C: Approved Random Number Generators for FIPS PUB 140-2, Security Requirements for Cryptographic Modules," *NIST Federal Information Processing Standards (FIPS) Publications*, 2012.

[12] F. Granelli and G. Boato, "A Novel Methodology For Analysis of The Computational Complexity of Block Ciphers: Rijndael, Camellia And Shacal-2 Compared," TS DIT-04-004, Department of Information And Communication Technology, University of Trento, 2004.

[13] D. He, C. Chen, S. Chan and J. Bu, "Analysis and improvement of a secure and efficient handover authentication for wireless networks," *Communications Letters, IEEE*, vol. 16, no. 8, pp. 1270–1273, 2012.

[14] D. He, C. Chen, S. Chan and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 48–53, 2012.

[15] A. Herzberg, H. Krawczyk and G. Tsudik, "On travelling incognito," in *1994 First IEEE Workshop on Mobile Computing Systems and Applications*, pp. 205–211, 1994.

[16] Q. Jiang, J. Ma, G. Li and L. Yang, "An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 68, no. 4, pp. 1477–1491, 2013.

[17] C. Kaufman, P. Hoffman, Y. Nir and P. Eronen, "Internet key exchange protocol version 2 (IKEv2)," *The Internet Engineering Task Force Request for Comments (IETF RFC)*, vol. 5996, 2010.

[18] S. S. Keller, "NIST-Recommended Random Number Generator Based on ANSI X9. 31 Appendix A. 2.4 Using the 3-key Triple DES and AES Algorithms," *National Institute of Standards an Technology (NIST)*, 2005.

[19] H. Liu and M. Liang, "Privacy-preserving registration protocol for mobile network," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1653–1671, 2014.

[20] P. Raj, "Highlights on telecom subscription data as on 30th june 2012," Press Release 171/2012, Telecom Regulatory Authority of India, 2012.

[21] G. Wedel and V. Kessler, "Formal semantics for authentication logics," in *Computer Security (Esorics'96)*, pp. 219–241, Springer, 1996.

[22] C. C. Yang, K. H. Chu and Y. W. Yang, "3G and WLAN interworking security: Current status and key issues," *International Journal of Network Security*, vol. 2, no. 1, pp. 1–13, 2006.

[23] T. Zhou and J. Xu, "Provable secure authentication protocol with anonymity for roaming service in global mobility networks," *Computer Networks*, vol. 55, no. 1, pp. 205–213, 2011.

Hiten Choudhury is an Assistant Professor in the Dept. of Computer Science and Information Technology at Cotton College State University, Assam, India. He has a Ph.D in Computer Science & Engineering from Tezpur Central University, Assam, India. His areas of research interest include wireless network security and authentication protocols.

Basav Roychoudhury is an Associate Professor of Information Systems at Indian Institute of Management, Shillong. He has a Ph.D in Computer Science and Engineering from Tezpur University, Assam, India. He has worked in the area of computer network protocols and security, enterprise systems, etc.

Dilip Kr. Saikia received his BE, M. Tech and Ph. D degrees from Madras University, IIT Madras and IIT Kharagpur in 1981, 1983 and 1996 respectively. He is currently a professor at the National Institute of Technology Meghalaya, India and the director of the institute. His current areas of research interest include network security, sensor networks and malware detection.

On Two Kinds of Flaws in Some Server-aided Verification Schemes

Zhengjun Cao[1], Lihua Liu[2], and Olivier Markowitch[3]
(Corresponding author: Zhengjun Cao)

Department of Mathematics, Shanghai University[1]
No.99, Shangda Road, Shanghai, China
Department of Mathematics, Shanghai Maritime University[2]
No.1550, Haigang Ave, Pudong New District, Shanghai, China
Computer Sciences Department, Université Libre de Bruxelles[3]
Boulevard du Triomphe CP 212, 1050 Bruxelles, Belgique
(Email: caozhj@shu.edu.cn)
(Received Sept. 30, 2015; revised and accepted Dec. 7, 2015)

Abstract

At Asiacrypt'05, Girault and Lefranc introduced the primitive of server-aided verification (SAV). In the proposed model, the server is assumed to be untrusted but is supposed to not collude with the legitimate prover. At ProvSec'08, Wu et al. have generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two corresponding SAV signature schemes, SAV-BLS-1 and SAV-BLS-2. In this paper, we argue that the SAV-BLS-1 scheme is somewhat artificial because the computational gain in the scheme is at the expense of additional communication costs. This is a common flaw in most outsourcing computation proposals which have neglected the comparisons between the computational gain and the incurred communication costs. We show also that the SAV-BLS-2 scheme is insecure against collusion attacks. It is another common flaw to have the verifier delegate most computations to the server in a way that prevent the verifier to confirm that the returned values are really bound to the signer's public key.

Keywords: Collusion attack, moderate adversary, outsourcing computation, server-aided verification

1 Introduction

Digital signatures and authentication schemes are broadly used in modern chips. The problem of speeding up the prover's or the signer's computations has interested many researchers. At Asiacrypt'05, Girault and Lefranc [7] formally introduced the primitive of server-aided verification (SAV) in order to speed up the verification task of a signature scheme or an identification scheme. They assumed that the verifier has only small computation capabilities while having access to a more powerful, but untrusted server or, equivalently, to a trusted server via a non authenticated communication link.

In a server-aided verification scheme, two kinds of deviating provers should be considered: The cheater who does not know the private key and the legitimate prover who misbehaves in order to make possible some kind of repudiation. For example, in a SAV signature scheme, an illegitimate prover and the server may collaborate in order to let the verifier accept a fake signature. Therefore, a SAV scheme must resist to collusion attacks launched by the server and an illegitimate prover (who may be represented by the same entity).

At Eurocrypt'95, Lim and Lee [13] put forth a generic method based on the "randomization" of the verification equation. However, the equation is only known to the verifier. In 2002, Girault and Quisquater [8] suggested a new approach based on the hardness of factorization and the composite discrete logarithm problem. At TCC'05, Hohenberger and Lysyanskaya [9] considered that an auxiliary server is made of two untrusted softwares which are assumed not to communicate with each other. In 2006, Dijk et al. [6] presented some protocols to speed up fixed-base variable-exponent exponentiation and variable-base fixed-exponent exponentiation using an untrusted computational resource.

At Asiacrypt'05, Girault and Lefranc [7] formally introduced the primitive of server-aided verification. In the model, the server is assumed to be untrusted but without colluding with the legitimate prover. At ProvSec'08, Wu et al. [19] have generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two corresponding SAV signature schemes, SAV-BLS-1 and SAV-BLS-2. They claimed that the SAV-BLS-1 is existentially unforgeable against adap-

tive chosen message attacks, and that the SAV-BLS-2 is sound against collusion attacks launched by the signer and the server. In 2011, Wu et al. [20] have proposed two another SAV signature schemes, SAV-Waters-1 and SAV-Waters-2, based on Waters' scheme [17]. In 2013, Wu et al. [18] have proved that both signature schemes [20] are insecure against collusion attacks launched by the legitimate signer and the server.

Recently, Lee et al. [11] have investigated the problem of cloud server-aided computation for ElGamal elliptic curve cryptosystem. Liao and Hsiao [12] studied the problem of multi-servers aided verification using self-certified public keys for mobile clients. Liu et al. [15] have investigated the problem of identity-based server-aided decryption. Zhang and Sun [22] proposed an ID-based server-aided verification of short signature scheme without key escrow.

In 2013, Canard et al. [3] considered the method for generically transforming a given well-known secure instance of a cryptographic primitive into a secure server-aided version where the server may be corrupted by the adversary. Chow et al. [5] revisited the definition of the security of server-aided verification. In 2014, Canard, Devigne and Sanders [2] provided some efficient ways to delegate the computation of a pairing $e(A, B)$, depending on the status of A and B. Their protocols enable the limited device to verify the value received from the third party by computing one exponentiation. In 2015, Liu et al. [16] considered the problem of server-aided anonymous attribute-based authentication in cloud computing. Very recently, Xiang and Tang [21] have proposed some efficient outsourcing schemes for modular exponentiations with checkability against untrusted cloud servers. Hsien et al. [10, 14] presented two surveys of public auditing for secure data storage in cloud computing. The computation of bilinear pairing represents most of the computing cost when dealing with pairing-based cryptographic protocols. In recent, Chen et al. [4] have put forth a new outsourcing algorithm for bilinear pairings in two untrusted programs model.

In this paper, we argue that the SAV-BLS-1 scheme is artificial because the delegated computations do not represent the heavier computation part; the verification procedure should therefore be rather performed solely by the verifier himself. What appears to be a computational gain in the scheme is due to communication costs that are not taken into account in the analysis. Nevertheless these costs could be far more important than the claimed computational gain. Then we show also that the SAV-BLS-2 scheme is insecure against collusion attacks launched by the server and illegitimate provers. This attack is possible because the verifier delegates its computations to the server in a way that prevent the verifier to confirm that the returned values from an adversary are really bound to the signer's public key. More generally, we point out that most outsourcing computation proposals have neglected the comparisons between the computational gains and the incurred communication costs.

2 The Reasonable Assumptions on SAV Model

Girault and Lefranc [7] assumed that the verifier has only a small computation capability but has access to a more powerful while untrusted server or, equivalently, to a trusted server via an unauthenticated communication link. From the practical of point of view, it appears to be more reasonable to assume that the verifier has access to a trusted server via an unauthenticated communication link, rather than to a potentially malicious server via an authenticated communication link (see Figure 1). This assumption holds on the following considerations:

1) The server is usually set up by some public service agencies. It is trusted due to the credibility of these agencies.

2) The server is assumed to serve many users. It is unrealistic to construct so many authenticated communication links between the server and so many users (chips with small computation capability).

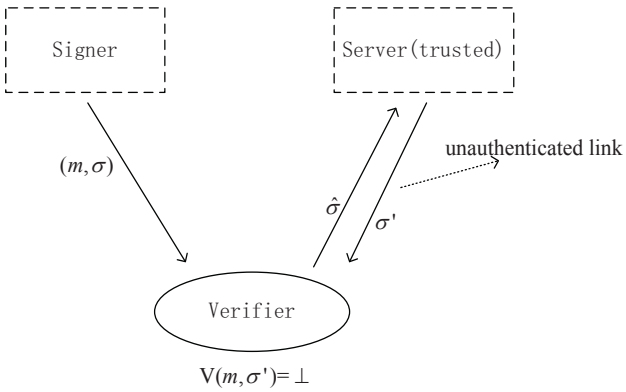


Figure 1: The trusted server with unauthenticated link in SAV model

3 Cryptanalysis of SAV-BLS-1 Signature Scheme

3.1 Review of SAV-BLS-1 Scheme

At ProvSec'08, Wu et al. generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two SAV signature schemes which are based on the BLS scheme [1]: SAV-BLS-1 and SAV-BLS-2. They claimed that the SAV-BLS-1 is existentially unforgeable against adaptive chosen message attacks, and that SAV-BLS-2 is sound against collusion attacks launched by the signer and the server.

The SAV-BLS-1 scheme can be described as follows.

1) ParamGen: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, for some prime number $p \geq 2^k$, k be the system security number and g be the generator of $\mathbb{G}_1$. e denotes the bilinear map $\mathbb{G}_1 \times$

Table 1: SA-Verify in SAV-BLS-1 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$r \in \mathbb{Z}_p$, $R = g^r$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{--\rightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$	$\overset{\sigma,R}{--\rightarrow}$	
		Compute $K_2 = e(H(m), y)^r$	$\overset{K_1}{\leftarrow --}$	$K_1 = e(\sigma, R)$
		Check $K_1 \overset{?}{=} K_2$		

$\mathbb{G}_1 \rightarrow \mathbb{G}_T$. There is one cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. The system parameter $param = (\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$.

2) KeyGen: The signer picks a random number $x \in \mathbb{Z}_p^*$ and keeps it as the secret key. The public key is set as $y = g^x$.

3) Sign: For a message m, the signer uses its secret key to generate the signature $\sigma = H(m)^x$.

4) Verify: For a message/signature pair (m, σ), one can check whether $e(\sigma, g) \overset{?}{=} e(H(m), y)$.

5) SA-Verifier-Setup: Given the system parameter $(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$, the verifier V randomly chooses $r \in \mathbb{Z}_p$ and sets $R = g^r$. The VString is (r, R).

6) SA-Verify: The verifier V and the server S interact with each other using the protocol described in Table 1. Note that R is precomputed and the verifier sends the same R to the server in server-aided verification of different message-signature pairs.

3.2 Analysis of SAV-BLS-1 Scheme

The authors claim that with the server's aid, if the parameters are properly selected, the verifier can save about half of the computational costs. However, we stress here that:

- Since the verifier has to compute $e(H(m), y)$ by himself, we are sure that the verifier has the capability to compute the delegated computation $e(\sigma, g)$. Therefore, the computational gain is only the cost for one pairing computation. According to actual effect, *the verifier is asking an equal capacity server for assistance, not a powerful server.*

- The verifier has to interact with the trusted server via a non authenticated link, by sending σ, R and receiving K_1. From the practical point of view, the communication costs (including authentication of the exchanged data, the possible underlying encryption/decryption, the time delay during the interaction, etc.) could be far more than the above computational gain (i.e., the cost of one pairing computation).

Based on these observations, we argue that the *direct verification* requires far less costs. Therefore, the SAV-BLS-1 signature scheme is somewhat unrealistic.

4 Cryptanalysis of SAV-BLS-2 Signature Scheme

4.1 Review of SAV-BLS-2 Scheme

The SAV-BLS-2 can be briefly described as follows.

- The phases of ParamGen, KeyGen, Sign and Verify are the same as that of SAV-BLS-1.

- SA-Verifier-Setup: Given the system parameter $(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$, the verifier computes $K_1 = e(g, g)$.

- SA-Verify: The verifier and the server interact with each other using the protocol described in Table 2.

4.2 Analysis of SAV-BLS-2 Scheme

In the SAV-BLS-2 scheme, the verifier delegates most of the computations to the server via the unauthenticated link. It is easy to find that the scheme is insecure against collusion attacks launched by the server (impersonated by an adversary) and an illegitimate prover, if the transferred data are not authenticated. In fact, the computations $\sigma' = \sigma g^r$, $K_2 = K_3 K_1^r$ performed by the verifier *do not invoke the public key* y. That means the verifier loses the ability to confirm that the returned values from the adversary are really bound to the signer's public key.

We describe here an attack launched by the server and an illegitimate prover: Let the illegitimate prover generate a fake signature $(m, H(m)^\theta)$ for some random $\theta \in \mathbb{Z}_p$ and send it to the verifier. Meanwhile, the prover sends θ to

Table 2: SA-Verify in SAV-BLS-2 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$K_1 = e(g, g)$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{\longrightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$		
		Pick $r \in \mathbb{Z}_p$, compute $\sigma' = \sigma g^r$	$\overset{m,\sigma',y}{\longrightarrow}$	$K_2 = e(\sigma', g)$
		Check $K_2 \overset{?}{=} K_3 K_1^r$	$\overset{K_2,K_3}{\longleftarrow}$	$K_3 = e(H(m), y)$

Table 3: An attack against SAV-BLS-2 signature scheme

Illegitimate prover		Verifier		Server (impersonated)
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y$.		$K_1 = e(g, g)$		
Given m, pick $\theta \in \mathbb{Z}_p$.	-----	----------	$\overset{\theta}{\longrightarrow}$	
Compute $\sigma = H(m)^\theta$	$\overset{m,\sigma,y}{\longrightarrow}$	Pick $r \in \mathbb{Z}_p$,	$\overset{m,\sigma',y}{\longrightarrow}$	$K_2 = e(\sigma', g)$
		compute $\sigma' = \sigma g^r$		
		Check $K_2 \overset{?}{=} K_3 K_1^r$	$\overset{K_2,K_3}{\longleftarrow}$	$K_3 = e(H(m), g^\theta)$

the server. See the following Table 3 for the details of the attack.

Correctness. It is easy to find that

$$\begin{aligned} K_3 K_1^r &= e(H(m), g^\theta) e(g, g)^r \\ &= e(H(m)^\theta, g) e(g^r, g) \\ &= e(H(m)^\theta g^r, g) \\ &= e(\sigma', g) = K_2 \end{aligned}$$

which means the verifier will accept the fake signature.

Remark 1. *If the verifying equation for a signature generated by a user is not bound to its public key, the verifier cannot be convinced that the signature is truly generated by the signer.*

Remark 2. *The adversary (who impersonates the server) himself can play the role of the illegitimate prover (see the modified definition of SAV model in Reference [19]). That means the SAV signature scheme is universally forgeable.*

5 Cryptanalysis of SAV-BLS-3 Signature Scheme

5.1 Review of SAV-BLS-3 Scheme

In 2011, Wu et al. [20] have proposed a new variation of SAV-BLS scheme. We briefly describe it as follows (see Table 4).

5.2 Analysis of SAV-BLS-3 Scheme

The scheme specifies that the computation of $e(H(m), y)$ is performed by the verifier himself. Thus the verification equation $K_2 = K_3^{r_1} K_1^{r_2}$ is really bounded to the signer's public key y and the challengers r_1, r_2 chosen by the verifier. In such case, our attack against the SAV-BLS-2 scheme fails. But we would like to stress that the SAV-BLS-3 scheme has the same flaw as the SAV-BLS-1 scheme. Namely, the verifier is asking an equal capacity server for assistance, not a powerful server.

It is easy to see that the SAV-BLS-3 scheme trades off the cost of one pairing computation for communication costs as well as the costs of some additional exponentiations. It is more inefficient than the SAV-BLS-1 scheme.

6 Further Siscussions

6.1 A Moderate Adversary

In 2013, Chow, Au and Susilo [5] modified the definition of the SAV security in [19, 20]. They clarified the goal of adversaries in the SAV model, and stressed that the adversary may benefit not only when an invalid signature is falsely-claimed as a valid one but also benefit from claiming a valid signature as invalid.

We now want to remark that a moderate adversary may try to make a fake signature but a wicked adversary

Table 4: SA-Verify in SAV-BLS-3 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$K_1 = e(g, g)$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{--\rightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$		
		Pick $r_1, r_2 \in \mathbb{Z}_p$, compute $\sigma' = \sigma^{r_1} g^{r_2}$	$\overset{\sigma'}{--\rightarrow}$	
			$\overset{K_2}{\leftarrow --}$	$K_2 = e(\sigma', g)$
		Compute $K_3 = e(H(m), y)$		
		Check $K_2 \overset{?}{=} K_3^{r_1} K_1^{r_2}$		

can ruin all valid signatures. A malicious server can always output some random values such that the verifier fails to check a valid signature. Of course, the latter is out the scope of academic study. From the practical point of view, it is reasonable to assume that the adversary is *moderate* at least: he may cheat the verifier in order to accept an invalid signature, instead of cheating the verifier to reject a valid signature. Otherwise, introducing a fully malicious adversary into the process of verification is meaningless.

Under this assumption, we think the most serious problem related to server-aided verification or outsourcing computation is when the computational gains are less than the incurred communication costs. We have observed that most outsourcing computation proposals had neglected the comparisons between the computational gains and the incurred costs. We argue that it is unnecessary for some SAV schemes to outsource one or two pairing computations at the expense of more communication costs if the verifier himself has the capability to compute the pairings.

6.2 A Nearby and Trusted Server

Girault and Lefranc [7] have described some situations in which a chip with a small computation capability is connected to a powerful device.

- In a GSM mobile telephone, the more sensitive cryptographic operations are performed in the so-called SIM (Subscriber Identification Module), which is already aided by the handset chip, mainly to decipher the over-the-air enciphered conversation.
- In a payment transaction, a so-called SAM (Secure Access Module) is embedded in a terminal already containing a more powerful chip.
- A smart card is plugged into a personal computer, seeing that many PCs will be equipped with smart card readers in a near future.

But we find that in all these situations (a SIM vs. a handset, a SAM vs. a powerful terminal, a smart card vs. a personal computer) the servers are nearby and trusted, not remote and untrusted.

In practice, we think, it is better to consider the scenario where a portable chip has access to a nearby and trusted server, but not to a remote server. Otherwise, the communication costs could overtake the computational gain of the outsourced computations.

7 Conclusion

In this paper we argue that the assumption on malicious server in the SAV model should be interpreted as a trusted server with unauthenticated channels, not an untrusted server with authenticated channels. We then argue that it incurs more communication costs which cannot be simply neglected and make most SAV schemes unpractical. We would like to stress that in a SAV signature scheme one has to balance carefully the delegated computations and the verifier's computations.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project 61303200, 61411146001) and the project of CRYPTASC (funded by the Brussels Institute for Research and Innovation). The authors gratefully acknowledge the reviewers for their valuable suggestions.

References

[1] D. Boneh, G. Lynn, and H. Shacham, "Short signature from the weil pairing," in *Proceedings of Advances in Cryptology (Asiacrypt'01)*, pp. 514–532, Gold Coast, Australia, December 2001.

[2] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *Proceedings of Applied Cryptography and Network Security*, pp. 549–565, Lausanne, Switzerland, June 2014.

[3] S. Canard and et al., "Toward generic method for server-aided cryptography," in *Proceedings of Information and Communications Security*, pp. 373–392, Beijing, China, November 2013.

[4] X. F. Chen and et al., "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, no. 562, pp. 112–121, 2015.

[5] S. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," *Information Security Technical Report*, vol. 17, pp. 46–57, 2013.

[6] M. Dijk and et al., "Speeding up exponentiation using an untrusted computational resource," *Designs, Codes and Cryptography*, vol. 39, pp. 253–273, 2006.

[7] M. Girault and D. Lefranc, "Server-aided verification: Theory and practice," in *Proceedings of Advances in Cryptology (Asiacrypt'05)*, pp. 605–623, Chennai, India, December 2005.

[8] M. Girault and J. Quisquater, "Gq + gps = new ideas + new protocols," in *Proceedings of Advances in Cryptology (Eurocrypt'02)*, Amsterdam, Netherlands, May 2002.

[9] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of Theory of Cryptography*, pp. 264–282, Cambridge, MA, USA, February 2005.

[10] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[11] N. Y. Lee, Z. L. Chen, and F. K. Chen, "Cloud server aided computation for elgamal elliptic curve cryptosystem," in *Proceedings of IEEE 37th Annual Workshops of Computer Software and Applications Conference*, pp. 11–15, Kyoto, Japan, July 2013.

[12] Y. P. Liao and C. M. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Computer Systems*, vol. 29, pp. 886–900, 2013.

[13] C. H. Lim and P. J. Lee, "Server (prover/signer)-aided verification of identity roofs and signatures," in *Proceedings of Advances in Cryptology (Eurocrypt'95)*, pp. 64–78, Saint-Malo, France, May 1995.

[14] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[15] J. Liu, C. K. Chu, and J. Y. Zhou, "Identity-based server-aided decryption," in *Proceedings of Information Security and Privacy (Acisp'11)*, pp. 337–352, Melbourne, Australia, July 2011.

[16] Z. S. Liu, H. Y. Yan, and Z. K. Li, "Server-aided anonymous attribute-based authentication in cloud computing," *Future Generation Computer Systems*, vol. 52, pp. 61–66, 2015.

[17] B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of Advances in Cryptology (Eurocrypt'05)*, pp. 114–127, Aarhus, Denmark, May 2005.

[18] H. Wu, C. X. Xu, J. Deng, and J. B. Ni, "On the security of two server-aided verification signature schemes," *Journal of Computational Information Systems*, vol. 9, no. 4, pp. 1449–1454, 2013.

[19] W. Wu, Y. Mu, W. Susilo, and X. Y. Huang, "Server-aided verification signatures: Definitions and new constructions," in *Proceedings of Provable Security*, pp. 141–155, Shanghai, China, November 2008.

[20] W. Wu, Y. Mu, W. Susilo, and X. Y. Huang, "Provably secure server-aided verification signatures," *Computers and Mathematics with Applications*, vol. 61, no. 7, pp. 1705–1723, 2011.

[21] C. Xiang and C. M. Tang, "Efficient outsourcing schemes of modular exponentiations with checkability for untrusted cloud server," *Journal of Ambient Intelligence and Humanized Computing*, no. 6, pp. 131–139, 2015.

[22] J. H. Zhang and Z. B. Sun, "An ID-based server-aided verification short signature scheme avoid key escrow," *Journal of Information Science and Engineering*, vol. 29, pp. 459–473, 2013.

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 Computer Sciences Department, Université Libre de Bruxelles, from 2008 to 2010. His 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. Her research interests include combinatorics, cryptography and information security.

Olivier Markowitch is an associate professor of the Computer Sciences Department at the Université Libre de Bruxelles. He is also information security advisor of his University. He is working on the design and analysis of two-party and multi-party cryptographic protocols as well as on the design and analysis of digital signature schemes.

A Stubborn Security Model Based on Three-factor Authentication and Modified Public Key

Trung Thanh Ngo and Tae-Young Choe
(Corresponding author: Tae-Young Choe)

Department of Computer Engineering, Kumoh National Institute of Technology
61 Daehak-ro, Gumi-si, Gyeongsangbuk-do, 730-701, Republic of Korea
(Email: choety@kumoh.ac.kr)
(Received Aug. 15, 2015; revised and accepted Nov. 27, 2015)

Abstract

Single authentication methods such as password, smart card, or biometric authentication suffer from their own weaknesses. Thus, combined authentication methods have been proposed recently. Unfortunately, even combined authentication methods are exposed to replay attacks, user impersonation attacks, server masquerading attacks, or stolen smart card attacks. To minimize the range of such attacks, we propose a security model that combines smart card authentication and biometric authentication using a modified public key cryptography. The modified public key cryptography transfers a public key only to the opposite entity not to public. The proposed security model can withstand the above-mentioned attacks. In particular, the insider attack can be resisted even in cases where the secret values stored in any two of three parties of a system are compromised. Such tolerance is enabled by modified public keys which are not revealed to the third party.

Keywords: Biometric authentication, modified public key signature, smart card, three-factor authentication, three-entity security model

1 Introduction

The rapid growth in cloud computing has recently offered many benefits to customers, such as large-scale computations and data storage, virtualization, high expansibility, high-reliability, and low service costs. Although cloud computing offers several benefits, it has indisputably given rise to many security issues; this is because a customer's data is stored on a cloud for access and processing. Among these issues, the authentication between a client and server is very critical because an intruder can break into a cloud system, and steal or modify a customer's sensitive data if the authentication scheme is weak.

Traditional remote identity authentication schemes mainly rely on the use of passwords between clients and servers [7, 8, 14]. However, simple passwords have been revealed easily through simple dictionary attacks.

To overcome this problem, a smart card and a password are used together to verify the identity of a user [1, 2, 4, 10, 11, 16, 17, 18, 19, 20]. A problem with this method is that the scheme is unable to provide non-repudiation owing to the fact that: Smart cards and passwords can be lost, forgotten, or even shared with other people. Therefore, it is impossible to determine the actual owner.

Fortunately, owing to the personal identity property of biometric keys (fingerprints, irises, hand geometry, faces, and so on), biometric verification can be included to provide non-repudiation. The advantages of using the additional properties of the biometric keys described in [11] are as follows:

- Biometric keys cannot be lost or forgotten.
- Biometric keys are very difficult to copy or share.
- Biometric keys are extremely hard to forge or distribute.
- Biometric keys cannot be guessed easily.
- The biometrics of one individual is not easier to break than that of another individual.

In combination with smart card and password verification, biometric verification makes remote authentication more secure and reliable. As a result, several researches have been conducted along this direction [3, 11, 12, 13].

Li and Hwang proposed an efficient biometric-based remote user authentication scheme using smart cards [11]. They claimed that besides maintaining good properties, such as unneeded synchronized clock, easy to change passwords, low computation costs, and mutual authentication

their method also provides non-repudiation because of the use of personal biometrics [14]. However, Das showed that Li and Hwang's scheme retains flaws in its login, authentication and password change phases, as well as in the verification of biometrics using a hash function [6]. Das proposed an improved scheme that resolves the flaws inherent to Li and Hwang's scheme. It maintains the scheme's good properties, and provides strong security against various attacks such as user impersonation attacks, server masquerading attacks, parallel session attacks, and stolen password attacks.

However, we found that some security flaws persist in Das scheme as well. In the case where the secret information stored in a smart card or on a server is lost, an attacker can conduct various kinds of attacks, such as a password change attack, server masquerading attack, or user impersonation attack. In this paper, we propose a new remote authentication scheme that improves on Das scheme, and shows how the new scheme can withstand the above-mentioned flaws. The proposed scheme sets the relations among the three entities: A biometric server, an authentication server, and a smart card representing a user. Each entity has single-side secrets to prevent an attack from dominating the system by leaking the secrets of the entities. In order to implement the single-side secrets, traditional public key distribution is modified. An entity has a private key and corresponding public key is assigned only to the pre-defined target entity. Although a message is encrypted with the private key, only the target entity can decrypt the message. Such modified public key system enables to detect whether the target entity is impersonated or not.

The rest of this paper is organized as follows. In Section 2, we briefly review Das scheme. The design flaws of Das scheme are discussed in Section 3. Section 4 presents our new scheme, which withstands the flaws discussed in Section 3. Next, the strength of our scheme is discussed in Section 5. Finally, we provide some concluding remarks in Section 6.

2 Review of Das Scheme

In Das scheme, there are four phases:

- Registration phase;
- Login phase;
- Authentication phase;
- Password change phase.

Notations used in this paper are shown in Table 1.

2.1 Registration Phase

To register with a trusted registration center TRC, a user must conform to the following steps:

1) User U_i provides TRC with their personal biometrics identity B_i and password PW_i personally.
2) Then, TRC computes the following:
   - $f_i \leftarrow h\ (B_i)$;
   - $r_i \leftarrow h\ (PW_i) \oplus f_i$;
   - $e_i \leftarrow h\ (ID_i \parallel X_i) \oplus r_i$.
3) TRC stores ID_i, $h(.)$, f_i, e_i, and r_i into the smart card SM_i and sends the smart card to the user in person.

2.2 Login Phase

To log in to the system, the user must adhere to the following 6 steps:

1) User U_i inserts a smart card into a card reader and offers her/his biometrics identity B_i' on a specific device for verification.
2) Biometric identity B_i' is matched against the biometrics template B_i of the user stored in the system.
3) If B_i' matches successfully, U_i passes the biometrics verification step and continues to Step 4. Otherwise, abort the remote authentication.
4) User U_i inputs password PW_i. The smart card SM_i computes $r_i' = h(PW_i) \oplus f_i$ and compares the result with r_i.
   - If $r_i = r_i'$, continue Step 5.
   - Otherwise, terminate the remote authentication.
5) The smart card SM_i computes the following:
   - $M_1 \leftarrow e_i \oplus r_i' = h(ID_i \parallel X_i)$;
   - $M_2 \leftarrow M_1 \oplus R_u = h(ID_i \parallel X_i) \oplus R_u$;
   - $M_3 \leftarrow h(R_u)$.
6) User U_i sends the message (ID_i, M_2, M_3) to S.

2.3 Authentication Phase

After receiving (ID_i, M_2, M_3) from U_i, server S processes the following ten steps:

1) Server S checks the format of the message.
   - If it is valid, continue to Step 2.
   - Otherwise, abort the login request.
2) Server S computes the following:
   - $M_4 \leftarrow h(ID_i \parallel X_i)$;
   - $M_5 \leftarrow M_2 \oplus M_4 = R_u$.
3) Server S verifies whether $M_3 = h(M_5)$.

- If they are equal, continue to Step 4.
- Otherwise, abort the login request.

4) Server S computes the following:
   - $M_6 \leftarrow M_4 \oplus R_s$;
   - $M_7 \leftarrow h(M_2 \parallel M_5)$;
   - $M_8 \leftarrow h(R_s)$.
5) Server S sends (M_6, M_7, M_8) to U_i.
6) After receiving (M_6, M_7, M_8), U_i verifies whether $M_7 = h(M_2 \parallel R_u)$.
   - If they are equal, continue to Step 7.
   - Otherwise, abort the login request.
7) U_i computes: $M_9 \leftarrow M_6 \oplus M_1$.
8) U_i verifies whether $M_8 = h(M_9)$.
   - If they are equal, U_i computes $M_{10} \leftarrow h(M_6 \parallel M_9)$.
   - Otherwise, abort the login request.
9) U_i sends (M_{10}) to S.
10) After receiving (M_{10}), S verifies whether $M_{10} = h(M_6 \parallel R_s)$.
    - If they are equal, S accepts the login request.
    - Otherwise, abort the login request.

2.4 Password Change Phase

To change their password, the user must conduct the following six steps:

1) U_i inserts a smart card into a card reader and offers her/his biometric identity B_i' on a specific device for verification.
2) B_i' is matched against the biometric template B_i of the user stored in the system.
   - If B_i' is valid, continue to Step 3.
   - Otherwise, abort the password change request.
3) U_i inputs the old password $PW_i{}^{old}$ and a new password $PW_i{}^{new}$.
4) The smart card SM_i computes the following:
   - $r_i' \leftarrow h(PW_i{}^{old}) \oplus f_i$;
   - If $r_i = r_i'$, continue to Step 5;
   - Otherwise, abort the change password request.
5) The smart card SM_i computes the following:
   - $r_i'' \leftarrow h(PW_i{}^{new}) \oplus f_i$;
   - $e_i' \leftarrow e_i \oplus r_i' = h(ID_i \parallel X_i)$;
   - $e_i'' \leftarrow e_i' \oplus r_i''$.
6) The smart card SM_i replaces e_i and r_i with e_i'' and r_i'', respectively.

3 Security Analysis of Das Scheme

In this section, we analyze the security of Das scheme based on the assumption that one of the following conditions is satisfied:

- An attacker can obtain all secret values of a smart card using a specific device to monitor the power consumption if they have one [5].
- An attacker can obtain all secret values of the server with the help of an insider of the server.
- An attacker can eavesdrop, intercept and modify messages sent between a user and server.

Under these assumptions, we analyze the following critical attacks on Das scheme.

3.1 Stolen Smart Card Attack

If a smart card is stolen, there is a possibility that an attacker can extract all secret values ID_i, $h(.)$, f_i, e_i, and r_i stored on the card using a specific device to monitor the power consumption [5]. With these values, the attacker can easily impersonate a legal user, as explained in Section 3.4 or masquerade as the server by forging authentication messages, as described in Section 3.5. The attacker can even guess or change the password of the smart card by conducting a password guessing attack, as described in Section 3.2 and a password changing attack, as detailed in Section 3.3.

3.2 Password Guessing Attack

The password can also be guessed using the following steps:

1) Attacker A uses secret values on the smart card to computes: $r_i \oplus f_i = h(PW_i)$.
2) A guesses password PW_i' and repeatedly verifies whether $h(PW_i') = h(PW_i)$ until the equation is satisfied.

A dictionary attack speeds up the guessing process [15].

3.3 Password Change Attack

Assume that legal user A_j picks up user U_i's smart card, and becomes an attacker. Attacker A_j can impersonate U_i and change the password in the smart card using her/his biometric information B_j. This can occur because there is no relationship between the biometrics and password verification processes. In addition, the password of the smart card can be broken using a simple dictionary attack. A dictionary attack is possible using the local password verification of the smart card, as shown in Step 4 of Section 2.2, and in Section 3.2.

In Li and Hwang's scheme [11], there is a relationship between the biometrics and password verification processes. After the biometrics verification, at the beginning of the login phase, it checks whether $f_i = h(B_i')$ where B_i' is the biometrics template inputted by the user. Therefore, if another legal user steals a smart card, the hashed value of their biometric template B_j does not match the value of f_i stored in the smart card, and they cannot use the smart card to log in to the system. Unfortunately, this step was discarded in Das scheme [6], because of the flaws in the biometrics verification using a hash function. As the result, Das scheme is exposed to the password change attack easily.

3.4 User Impersonation Attack

If attacker A knows all secret values $(ID_i, h(.), f_i, e_i, r_i)$ of user U_i stored in a smart card, A can easily impersonate U_i through the following steps:

1) Attacker A generates and sends a message (ID_i, M_2^f, M_3^f) to server S, where
   - $M_1 \leftarrow e_i \oplus r_i$;
   - $M_2^f \leftarrow M_1 \oplus R_f$;
   - $M_3^f \leftarrow h(R_f)$;
   - R_f is a random number generated by A.
2) Server S receives (ID_i, M_2^f, M_3^f) and checks whether ID_i is valid. Because ID_i is valid, S computes the following:
   - $M_4 \leftarrow h(ID_i \parallel X_i)$;
   - $M_5^f \leftarrow M_2^f \oplus M_4$.
3) Server S checks and sees that $M_3^f = h(M_5^f)$, and then sends (M_6, M_7^f, M_8) to A, where
   - $M_6 \leftarrow M_4 \oplus R_s$;
   - $M_7^f \leftarrow h(M_2^f \parallel M_5^f)$;
   - $M_8 \leftarrow h(R_s)$.
4) Attacker A receives the message (M_6, M_7^f, M_8) and computes the following:
   - $M_9 \leftarrow M_6^f \oplus M_1$;
   - $M_{10} \leftarrow h(M_6 \parallel M_9)$.
5) Attacker A sends M_{10} to S.
6) After receiving (M_{10}), S verifies whether $M_{10} = h(M_6 \parallel R_s)$.
7) Server S grants access to the attacker.

At this point, the attacker has successfully impersonated user U_i.

3.5 Server Masquerading Attack

If attacker A knows all the secret values stored in a smart card, A can easily masquerade as the server by conducting the following steps:

1) Attacker A sends a message (M_6^f, M_7, M_8^f) to U_i, where
   - $M_4 \leftarrow e_i \oplus r_i$;
   - $M_5 \leftarrow M_2 \oplus M_4$;
   - $M_6^f \leftarrow M_4 \oplus R_f$;
   - $M_7 \leftarrow h(M_2 \parallel M_5)$;
   - $M_8^f \leftarrow h(R_f)$;
   - R_f is a random number generated by the attacker.
2) User U_i receives (M_6^f, M_7, M_8^f), verifies that $M_7 = h(M_2 \parallel R_u)$, and computes M_9^f where, $M_9^f \leftarrow M_6^f \oplus M_1$.
3) User U_i verifies that $M_8^f = h(M_9^f)$, and then it computes and sends (M_{10}) to the attacker, where $M_{10} \leftarrow h(M_6^f \parallel M_9^f)$.

At this point, the attacker has successfully masqueraded as the server.

3.6 Insider Attack

If attacker A knows the secret value X_i of user U_i stored on the server with the help of an insider, attacker A can impersonate user U_i by conducting the following steps:

1) Attacker A sends a message (ID_i, M_2^f, M_3^f) to the server S, where
   - $M_2^f \leftarrow h(ID_i \parallel X_i) \oplus R_f$;
   - $M_3^f \leftarrow h(R_f)$;
   - R_f is a random number generated by the attacker.
2) Attacker A receives the reply message (M_6, M_7, M_8) from the server, computes M_{10}^f, and sends it to S, where
   - $M_9 \leftarrow M_6 \oplus h(ID_i \parallel X_i)$;
   - $M_{10}^f \leftarrow h(M_6 \parallel M_9)$.
3) Server S verifies that $M_{10}^f \leftarrow h(M_6 \parallel R_s)$. Attacker A then grants login access to server S.

Table 1: Notations

Notation	Description
U_i	User i
TRC	Trusted Registration Center
SM_i	Smart card of user i
S	Server
A	Attacker
PW_i	Password of the user i
ID_i	Identity of the user i
B_i	Biometric template of the user i
$h(.)$	A secure hash function
X_i	A secret information for user U_i maintained by the server S
$a \parallel b$	a concatenates b
$a \oplus b$	a Exclusive-OR b
R_u	A random number generated by the user U_i
R_s	A random number generated by the server S
R_{bs}	A random number generated by the biometric server BS
$\{M\}_U$	Encrypts message M using public key of user U
$[M]_U$	Encrypts message M using private key of user U

4 Enhanced Scheme

In this section, we propose an enhanced scheme that overcomes the flaws of Das scheme and resists the attacks discussed in Section 3.

To begin with, we describe the system model used in the enhanced scheme. Das scheme uses a client-server model consisting of a remote server and many clients. The client side includes a terminal, smart card reader, and local biometric verification system that takes the user's biometric template using a specific device and compares it with the user's biometric template that is stored in a local database.

However, the local biometrics verification system is not scalable for cloud computing systems because a local biometrics database cannot store the biometrics templates of innumerable cloud users. Therefore, biometrics verification must be processed by remote servers.

In our scheme, the system model consists of smart card SM_i, a biometrics server BS, and an authentication server S, as depicted in the Figure 1.

Further, we improve the security using public key cryptography along with random nonces. The public and private key pairs are distributed securely as depicted in Figure 1. These key pairs are used for the encryption and mutual authentication in each phase of the scheme. The public and private keys are used differently than a general public/private key pair. In general, a public key is freely accessible, and is used for encryption. However, our public key is only a partially secure key used to encrypt a users' plaintext and generate a ciphertext. The corresponding private key is used to decrypt the ciphertext. For example, user U generates private key PR_U and public key PB_U. U sends PB_U to another user W securely. If U sends an encrypted message $[M]_U$, only W can decrypt it using PB_U and W knows that M is sent from U. In addition, if W sends an encrypted message M_U, only U can decrypt the message using PR_U and know that N was sent from W. One thing we must remember is that a standard such as $PKCS$ by RSA Security Inc. cannot be used because a public key is included in a private key format [9]. The enhanced scheme consists of four phases which are Registration phase, Login phase, Biometrics reporting phase, and Server authentication phase. In order to access the server, the user must follow these phases sequentially.

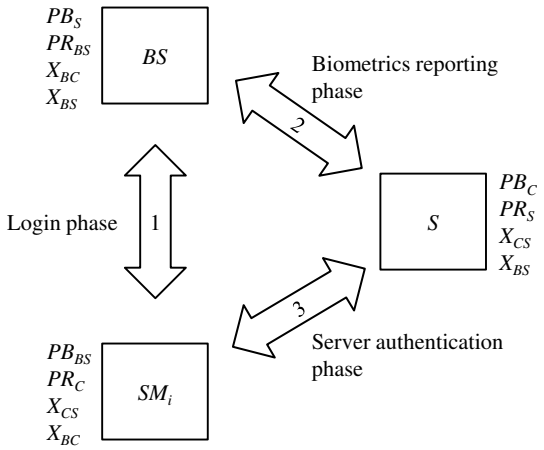


Figure 1: Authentication model in three entities

4.1 Registration Phase

To log in to the system, the user must visit a registration center in person and register for an account. At the regis-

tration center, the user and center perform the following steps:

1) The trusted registration center TRC generates the following keys:
   - Private/public key pair PR_{BS}/PB_{BS} for use between BS and SM_i.
   - Private/public key pair PR_C/PB_C for use between SM_i and S.
   - Private/Public key pair PR_S/PB_S for use between S and BS.
   - Secret Key X_{BC} is shared between BS and SM_i.
   - Secret Key X_{CS} is shared between S and SM_i.
   - Secret Key X_{BS} is shared between BS and S.
2) The user inputs their biometric identity B_i on a specific device and offers their password PW_i.
3) TRC computes the following:
   - $f_i \leftarrow h(B_i)$;
   - $r_i \leftarrow h(PW_i) \oplus f_i$;
   - $e_i \leftarrow h(ID_i \parallel X_{CS}) \oplus r_i$.
4) TRC stores ID_i, $h(.)$, e_i, r_i, X_{BC}, X_{CS}, PB_B, and PR_C into a smart card SM_i and delivers the smart card to user U_i in person.
5) Finally, TRC distributes the remaining keys and information through a secure channel:
   - $PR_{BS}, PB_S, X_{BC}, X_{BS}, B_i$, and ID_i to biometric server BS.
   - $PR_S, PB_C, X_{CS}, X_{BS}, ID_i, PW_i$, and $h(ID_i \parallel h(PW_i))$ to authentication server S.

4.2 Login Phase

After registering for an account, user U_i can log in to the system by performing the following steps:

1) U_i inserts a smart card SM_i into a card reader and inputs their biometrics identity B_i' on a biometrics reading device.
2) The smart card SM_i uses the public key PB_{BS} to encrypt ID_i, and sends $\{ID_i\}_{BS}$ to the biometrics server BS.
3) When BS receives the message from SM_i, it uses the private key PR_{BS} to decrypt the message to obtain ID_i.
4) BS computes the following messages:
   - $M_1 \leftarrow h(ID_i \parallel X_{BC}) \oplus R_{bs}$;
   - $M_2 \leftarrow h(R_{bs})$;
   - R_{bs} is a random number generated by BS.
5) BS uses the private key PR_{BS} to encrypt and sends the message $[(M_1, M_2)]_{BS}$ to SM_i. Because its corresponding public key is not revealed to the public, the encrypted message is transferred securely. Simultaneously, the sender is authenticated as BS.
6) When SM_i receives $[(M_1, M_2)]_{BS}$ from BS, it uses the public key PB_{BS} to decrypt the message and then computes the following messages:
   - $M_3 \leftarrow h(ID_i \parallel X_{BC})$;
   - $M_4 \leftarrow M_1 \oplus M_3$.
7) SM_i verifies whether $h(M_4) = M_2$:
   - If this is not true, SM_i terminates the login phase.
   - Otherwise, the process continues to Step 8.
8) SM_i computes M_5 and sends a message $\{(M_5, B_i')\}_{BS}$ encrypted with the public key PB_{BS} to BS, where $M_5 \leftarrow h(M_1||M_4)$.
9) After receiving $\{(M_5, B_i')\}_{BS}$, BS uses its private key PR_{BS} to decrypt the message and then verifies whether $M_5 = h(M_1 \parallel R_{bs})$
   - If these are equals, BS compares B_i' with the biometric template B_i of user ID_i in the database.
     - If B_i' is valid, BS accepts the login request of SM_i, sends an acceptance message to SM_i and moves on to the biometrics report phase.
     - Otherwise, BS rejects the login request of SM_i.
   - Otherwise, BS rejects the login request of SM_i.

The login phase is summarized in Login phase box of Figure 2.

4.3 Biometrics Reporting Phase

After accepting the login request, the biometrics server BS must report the login result to server S by performing the following steps:

1) BS uses the public key PB_S to encrypt the user's id ID_i obtained in the login phase and sends $\{(ID_i, R_{bs})\}_S$ to S, where R_{bs} is a random number generated by BS.
2) S receives $\{(ID_i, R_{bs})\}_S$ and uses the private key PR_S to decrypt it.
3) S computes the following messages:
   - $M_1 \leftarrow h(ID_i \parallel X_{BS}) \oplus R_{ss}$;
   - $M_2 \leftarrow h(R_{ss})$;

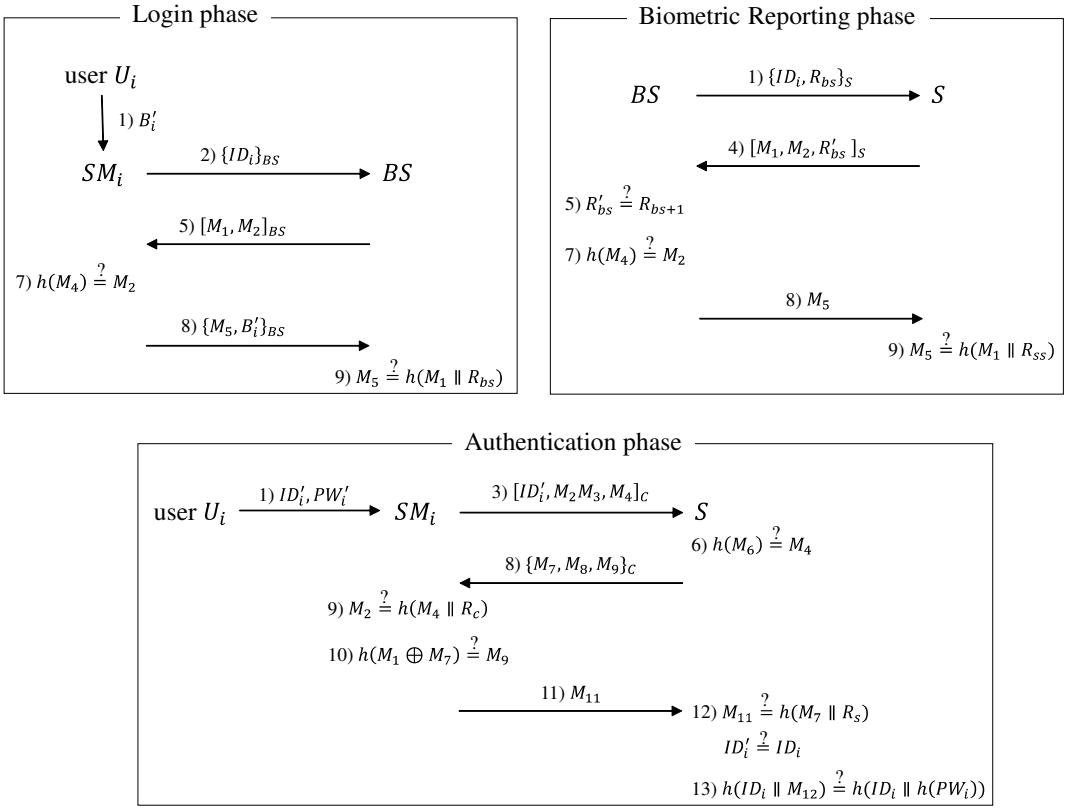


Figure 2: Three phases for remote user authentication

- R_{ss} is a random number generated by S;
- $R_{bs}' \leftarrow R_{bs} + 1$.

4) S uses the private key PR_S to encrypt and sends the message $[(M_1, M_2, R_{bs}')]_S$ to BS to authenticate BS.

5) When BS receives $[(M_1, M_2, R_{bs}')]_S$ from S, it uses the public key PB_S to decrypt the message and then verifies whether $R_{bs}' = R_{bs} + 1$.

- If this is false, BS terminates the biometrics reporting phase.
- Otherwise, the process continues to Step 6.

6) BS computes the following messages:

- $M_3 \leftarrow h(ID_i \| X_{BS})$;
- $M_4 \leftarrow M_1 \oplus M_3$.

7) BS verifies whether $h(M_4) = M_2$.

- If this is false, BS terminates the biometrics reporting phase.
- Otherwise, the process continues to Step 8.

8) BS computes M_5 and sends it to S, where $M_5 \leftarrow h(M_1||M_4) = h((h(ID_i \| X_{BS}) \oplus R_{ss}) ||R_{ss})$.

9) After receiving M_5, S verifies whether $M_5 = h(M_1 \| R_{ss})$.

- If this is true, move on to the authentication phase.
- Otherwise, S terminates the session.

The summary of the biometric reporting phase is shown in Biometric Reporting phase box of Figure 2.

4.4 Authentication Phase

After receiving the acceptance message from BS, server S authenticates user U_i by conducting the following steps:

1) U_i inputs their password PW_i' into the smart card SM_i.

2) SM_i computes the following messages:

- $r_i' \leftarrow h(PW_i')$;
- $M_1 \leftarrow h(ID_i' \oplus X_{CS})$;
- $M_2 \leftarrow M_1 \oplus R_c$;
- $M_3 \leftarrow r_i' \oplus R_c$;
- $M_4 \leftarrow h(R_c)$;
- R_c is a random number generated by SM_i.

3) SM_i uses the private key PR_C to encrypt and sends the message $[(ID_i', M_2, M_3, M_4)]_C$ to server S. PR_C can be used to encrypt the message is because its corresponding public key PB_C is not revealed to the public. PB_C is stored only in the server S.

4) S receives the message and uses the public key PB_C to decrypt it to obtain (ID_i', M_2, M_3, M_4).

5) S computes the following messages:

   - $M_5 \leftarrow h(ID_i' \oplus X_{CS})$;
   - $M_6 \leftarrow M_5 \oplus M_2$.

6) S verifies whether $h(M_6) = M_4$. If this is false, S terminates the session. Otherwise, the process continues to Step 7.

7) S computes the following messages:

   - $M_7 \leftarrow M_5 \oplus R_s$;
   - $M_8 \leftarrow h(M_2 \parallel M_6)$;
   - $M_9 \leftarrow h(R_s)$;
   - R_s is a random number generated by S.

8) S uses the public key PB_C to encrypt and sends the message $\{(M_7, M_8, M_9)\}_C$ to SM_i.

9) SM_i receives $\{(M_7, M_8, M_9)\}_C$, and uses the private key PR_C to decrypt it. Then SM_i verifies whether $M_8 = h(M_2||R_c)$.

10) If this is false, SM_i terminates the session. Otherwise, SM_i verifies whether $h(M_{10}) = M_9$ where $M_{10} \leftarrow M_1 \oplus M_7$.

11) If this is false, SM_i terminates the session. Otherwise, SM_i computes and sends M_{11} to S where $M_{11} \leftarrow h(M_7 \parallel M_{10})$.

12) S receives M_{11} and verifies whether $M_{11} = h(M_7 \parallel R_s)$.

    - If this is false, S terminates the session.
    - Otherwise, S compares ID_i received from the biometrics server during the biometrics reporting phase with ID_i' received from the smart card in Step 4. If the two values of ID_i' do not match, S terminates the session. Otherwise, the process moves to Step 13.

13) S computes $h(ID_i \parallel M_{12})$ and compares the result with $h(ID_i \parallel h(PW_i))$ in the database, where $M_{12} \leftarrow M_3 \oplus M_6$.

    - If this is true, S accepts the login request.
    - Otherwise, S rejects the login request.

In this phase, a mutual authentication process between smart card SM_i and server S is used along with public key cryptography and random nonces. From Steps 1 through 3, smart card SM_i computes, encrypts, and sends messages containing its random nonce R_c to server S. From Steps 4 through 8, server S verifies the random nonce R_c, and computes, encrypts, and sends reply messages containing its random nonce R_s to smart card SM_i. From Steps 9 through 11, smart card SM_i verifies the random nonce R_s and, replies to server S. Finally, server S verifies the last reply message from SM_i, and conducts the smart card password verification in Steps 12 and 13, respectively. The authentication phase is summarized in Authentication phase box of Figure 2.

5 Security Analysis of the Enhanced Scheme

In this section, we analyze the security of the enhanced scheme based on the assumptions stated in Section 3 and show that our scheme can withstand all the mentioned attacks on Das scheme.

5.1 User Impersonation Attack

As mentioned in Section 3.1, Das scheme can be attacked easily if an attacker knows all the secret values stored in a user's smart card. However, this attack cannot succeed in our scheme. When attacker A tries to send the message $[(ID_i, M_2, M_3, M_4)]_C$ to S during the authentication phase, A uses the secret values of the smart card to compute fake messages M_2 and M_3, where

- $M_1 \leftarrow e_i \oplus r_i = h(ID_i \oplus X_{CS})$;
- $M_2 \leftarrow M_1 \oplus R_a$;
- $M_4 \leftarrow h(R_a)$;
- R_a is a random number generated by an attacker.

However, attacker A cannot generate M_3 because $M_3 = h(PW_i') \oplus R_c$ where PW_i' is not stolen by attacker A. In addition, attacker A cannot obtain $h(PW_i')$ after intercepting $[(ID_i, M_2, M_3, M_4)]_C$ because PB_C is needed to decrypt the message.

5.2 Server Masquerading Attack

As mentioned in Section 3.2, Das scheme can be attacked easily if an attacker knows all the secret values stored in a user's smart card. However, such an attack cannot work in our scheme because an attacker does not know the server's public key PB_C used to decrypt the message $[(ID_i, M_2, M_3, M_4)]_C$.

5.3 Password Guessing Attack

Unlike Das scheme, our proposed scheme does not store the secret value $f_i = h(B_i)$ into a smart card during the registration phase. Therefore, it is impossible for an attacker to execute this type of an attack.

5.4 Password Changing Attack

As pointed out in Section 3.3, there is no relationship between the biometrics and password verification processes in Das scheme. Thus, if attacker A obtains smart card SM_i of user U_i, A can impersonate U_i using U_i's biometric information.

In our scheme, however, this attack is impossible because after passing the login phase, the biometrics server reports the login user's id to the server in the biometrics reporting phase, as described in Section 4.3. Therefore, the server knows who has passed the login phase and is able to verify whether the smart card belongs to that person. This is conducted in the Steps 12 and 13 of the server authentication phase.

Moreover, to prevent this kind of attack, our scheme does not allow the user to change their smart card's password locally. If a user loses or forgets their smart card's password, they must contact the registration center to retrieve the password or create a new account.

5.5 Insider Attack

This attack is based on the assumption with the help of an insider, an attacker can acquire all the secret values stored in biometrics server BS, the server S or both. This assumption and the previous assumption on a loss of the smart card's secret values, motivated us to design a scheme that remains secure despite losing all secret values stored in any one or two parties of a system.

This expectation has been achieved in our scheme using public key cryptography. To demonstrate this, we consider the following cases in which secret values are lost.

Case 1: All secret values stored in BS and SM_i are compromised. Attacker A can pass the login and biometrics reporting phase, but not the server authentication phase. During the login phase, an attacker can easily impersonate a legal user by conducting the following steps:

1) Attacker A eavesdrops on an encrypted message $\{(M_5, B_i')\}_{BS}$ sent from SM_i to BS in Step 2 of the login phase (Section 4.1).

2) Attacker A uses the private key PR_{BS} to decrypt the message to obtain B_i'.

3) Later, A starts the login phase by sending message $\{ID_i\}_{BS}$ to BS to impersonate SM_i.

4) BS uses private key PR_{BS} to decrypt the message to obtain ID_i, and sends the message $[(M_1, M_2)]_{BS}$ back to A.

   - $M_1 \leftarrow h(ID_i \parallel X_{BC}) \oplus R_{bs}$;
   - $M_2 \leftarrow h(R_{bs})$;
   - R_{bs} is a random number generated by BS.

5) Attacker A uses the public key PB_{BS} to decrypt $[(M_1, M_2)]_{BS}$, forges M_5, and sends message $\{(M_5, B_i')\}_{BS}$ encrypted with public key PB_{BS} to BS, where

   - $M_3 \leftarrow h(ID_i \parallel X_{BC})$;
   - $M_4 \leftarrow M_1 \oplus M_3$;
   - $M_5 \leftarrow h(M_1 \parallel M_4) = h((h(ID_i \parallel X_{BC}) \oplus R_a) \parallel R_a)$;
   - R_a is a random number generated by A.

6) Biometric server BS verifies that M_5 and B_i are valid. BS then allows A to proceed to the server authentication phase. Next, BS automatically proceeds to the biometrics reporting phase with the ID_i provided by the attacker. Therefore, the attacker does not need to carry out an attack on the biometrics reporting phase.

In the server authentication phase, as explained in the Section 5.1, the attacker cannot impersonate the user if he has only the secret values of the user's smart card SM_i without password PW_i.

Case 2: All secret values stored in BS and S are compromised. Attacker A can pass only the biometrics reporting phase, but not the login phase or the server authentication phase.

During the biometrics reporting phase, the attacker A does not need to impersonate BS because the phase is processed automatically by both BS and S.

During the login phase, in order to impersonate the user U_i, A must have the public key PB_{BS} to decrypt the message $[(M_1, M_2)]_{BS}$ sent by BS in Step 6 of Section 4.2. After decrypting the message, A can use M_1 to forge reply message M_5 to trick BS into recognizing him as a legal user. However, A does not know PB_{BS}, which is stored only in SM_i. Consequently, A fails to impersonate U_i.

During the authentication phase, attacker A intercepts message $[(ID_i', M_2, M_3, M_4)]_{PR_C}$, and forwards it to S between Steps 3 and 4 of the authentication phase in order to impersonate the user U_i. When A receives the message $\{(M_8, M_9, M_{10})\}_{PB_C}$ in Step 9, A cannot decrypt it because PR_C is stored in smart card SM_i. In addition, A cannot generate M_{12}, which must be authenticated by S.

Case 3: All secret values stored in SM_i and S are compromised. Attacker A can pass both the biometrics reporting phase and the authentication phase.

During the login phase (Section 4.2), attacker A cannot impersonate a legal user because A does not know the biometric identity B_i' of the user U_i sent to the biometric server BS in Step 8. Moreover, it is impossible for A to retrieve B_i' without knowing the private key PR_{BS} stored in the biometric server.

However, attacker A knows all the secret values stored in smart card SM_i and server S, and can easily forge fake

Table 2: Analysis results of the insider attack

Compromised Sites			Cracked Phases			Guard Values
SM	BS	S	**Login**	**Biometrics Reporting**	**Authentication**	
O	O	X	O	O	X	PB_C
X	O	O	X	O	X	PB_{BS}, PR_C
O	X	O	X	O	O	PR_{BS}

messages needed to impersonate a legal user during the authentication phase. In addition, A can bypass the password verification in Step 13 of the authentication phase by conducting the following steps:

- Attacker A eavesdrops on message $[(ID_i, M_2, M_3, M_4)]_C$ sent from smart card SM_i in Step 3 of the authentication phase.
- A uses the public key PB_C to decrypt the message.
- A computes the following:
  - $M_1 \leftarrow e_i \oplus r_i = h(ID_i \oplus X_{CS})$;
  - $R_c \leftarrow M_1 \oplus M_2$;
  - $r_i' \leftarrow M_3 \oplus R_c = h(PW_i')$.
- At this point, A can compute $h(ID_i \parallel h(PW_i'))$, and use it to pass Step 13 of the authentication phase.

During the biometrics reporting phase, attacker A does not need to impersonate BS because the phase is processed automatically by both BS and S. The Table 2 summarizes the analysis results of an insider attack.

6 Conclusion

In this paper, we reviewed and analyzed the security of Das scheme. We showed that this scheme still retains certain flaws that make it insecure against various types of attacks, particularly an insider attack. Therefore, we redesigned the system model and added the use of public key cryptography to overcome these security weaknesses, and made the system more secure against various types of attacks. For an insider attack, even when secret information stored in a smart card, in the biometrics server, or in the authentication server is revealed, an attacker still cannot pass the authentication process.

Acknowledgments

This research was supported by research fund, Kumoh National Institute of Technology.

References

[1] R. Amin, "Cryptanalysis and efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card," *International Journal of Network Security*, vol. 18, pp. 172–181, January 2016.

[2] R. Amin and G. P. Biswas, "An improved rsa based user authentication and session key agreement protocol usable in tmis," *Journal of Medical Systems*, vol. 39, no. 8, pp. 1–14, 2015.

[3] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for tmis with user anonymity," *Journal of medical systems*, vol. 39, no. 8, pp. 1–19, 2015.

[4] R. Amin, T. Maitra and S. P. Rana, "An improvement of wang. et. al.'s remote user authentication scheme against smart card security breach," *International Journal of Computer Applications*, vol. 75, no. 13, pp. 37–42, 2013.

[5] Y. An, "Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards," *BioMed Research International*, vol. 2012, 2012.

[6] A. K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[7] L. Fan, J. H. Li and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 7, pp. 665–667, 2002.

[8] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[9] J. Jonsson and B. Kaliski, "Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1," 2003. (`https://tools.ietf.org/html/rfc3447`) referenced at 3 March 2015.

[10] N. Y. Lee and Y. C. Chiu, "Improved remote authentication scheme with smart card," *Computer Standards & Interfaces*, vol. 27, no. 2, pp. 177–180, 2005.

[11] C. T. Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[12] X. Li, J. W. Niu, J. Ma, W. D. Wang and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.

[13] C. H. Lin and Y. Y. Lai, "A flexible biometrics remote user authentication scheme," *Computer Standards & Interfaces*, vol. 27, no. 1, pp. 19–23, 2004.

[14] J. J. Shen, C. W. Lin and M. S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Computers & Security*, vol. 22, no. 7, pp. 591–595, 2003.

[15] R. Shirey, "RFC 2828: Internet security glossary," 2000. (`http://tools.ietf.org/html/rfc2828`) referenced at 5 March 2015.

[16] S. K. Sood, A. K. Sarje and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[17] H. M. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958–961, 2000.

[18] D. Wang, C. G. Ma, Q. M. Zhang and S. Zhao, "Secure password-based remote user authentication scheme against smart card security breach," *Journal of Networks*, vol. 8, no. 1, pp. 148–155, 2013.

[19] S. T. Wu and B. C. Chieu, "A user friendly remote authentication scheme with smart cards," *Computers & Security*, vol. 22, no. 6, pp. 547–550, 2003.

[20] E. J. Yoon, E. K. Ryu and K. Y. Yoo, "Further improvement of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 612–614, 2004.

Trung Thanh Ngo received his bachelor's degree in Computer Science from Troy University, Alabama, USA in 2012. He received his MS degree in Computer Engineering from Kumoh National Institute of Technology, Gumi, South Korea in 2015. Currently, he is working as a system engineer for a telecommunication company in South Korea. His main research interest is secure authentication protocol in cloud system.

Tae-Young Choe received his bachelor's degree in Mathematical Education from Korea University, Seoul, South Korea in 1991. He received his MS and PhD degree in Computer Engineering from POSTECH, Pohang, South Korea in 1996 and 2002, respectively. Since 2002, he has been with KIT, Kumoh National Institute of Technology, Kumi, South Korea, where he is a professor in the Department of Computer Engineering. His main research interests are task management and secure authentication in cloud system.

Secret Share Based Program Access Authorization Protocol for Smart Metering

Xiuxia Tian[1], Lisha Li[2], Jinguo Li[1], Hongjiao Li[1] and Chunhua Gu[1]
(Corresponding author: Xiuxia Tian)
School of Computer Science and Technology, Shanghai University of Electric Power[1]
No. 2588 Changyang Road, Shanghai 200090, China
School of Electronics and Information Engineering, Shanghai University of Electric Power[2]
No. 2588 Changyang Road, Shanghai 200090, China
(Email: xxtian@fudan.edu.cn)
(Received Aug. 16, 2015; revised and accepted Nov. 27, 2015 & Jan. 3, 2016)

Abstract

Varieties of data security protection technologies have been developed in smart metering. However, there are almost no researches focusing on the security of smart meter parameters. In fact, smart meter parameters, such as the total/sharp/peak/flat/valley period time, can be written locally or remotely by power companies or users through sending programming commands. Modifying smart meter parameters arbitrarily may lead to the non-authenticity of users' energy bills. This paper first considers the security of smart meter parameters and proposes a novel secret share based program access authorization protocol. The protocol introduces a program lock key which is used to lock the smart meter parameter modifying program. At least t participants (i.e. one user and $t-1$ power companies) jointly can have the authorization to obtain the program lock key. We describe the security and efficiency analysis through theory and experiments.

Keywords: Program access, smart meter parameters, security protection, secret share

1 Introduction

Smart meters have been widely deployed all over the world. Varieties of security protection technologies have been developed in smart metering. Most of them focus on the smart meter data security protection. However, almost no approaches focus on the security of smart meter parameters. Smart meter parameters, such as the total/sharp/peak/flat/valley period time, should be protected from being modified arbitrarily because they affect the authenticity of users' energy bills.

As shown in Figure 1, we take the sharp/peak period time for example. Assume the peak period time is $3pm-7pm$ and the sharp period time is $7pm-10pm$; the electricity price is 1.21567 $yuan/kwh$ in the sharp period time and 1.14460 $yuan/kwh$ in the peak period time. If a valid user's monthly electricity consumption is 600 kwh in the sharp period time and 100 kwh in the peak period time, the real bill is 843.86200 $yuan$, as shown in Figure 1. Assume the valid user's monthly electricity consumption/hour at one period time is equal. Thus the valid user's monthly electricity consumption/hour is $600 \div 3 = 200$ kwh at the sharp period time and $100 \div 4 = 25$ kwh at the peak period time. A curious power company may modify the peak period time from $3pm-7pm$ to $3pm-6pm$ and in turn the sharp period time from $7pm-10pm$ to $6pm-10pm$ through sending programming commands. The energy bill is changed to 845.63875 $yuan$, as shown in Figure 1. Thus the valid user needs to pay $845.63875-843.86200=1.77675$ $yuan$ more than what he need to pay while the curious power company will acquire additional profit 1.77675 $yuan$. A malicious user may modify the peak period time from $3pm-7pm$ to $3pm-8pm$ and in turn the sharp period time from $7pm-10pm$ to $8pm-10pm$, which is a behavior of stealing electricity. The energy bill becomes 829.64800 $yuan$, as shown in Figure 1. So the malicious user may pay $843.86200-829.64800=14.21400$ $yuan$ less than the real bill while the valid power company will loss 14.21400 $yuan$.

Therefore, we need to limit the modifying permissions on smart meter parameters. By introducing a program lock key which is used to lock the smart meter parameter modifying program, we present a novel secret share based program access authorization protocol to solve the problem. Any one user or power company can not arbitrarily modify smart meter parameters for their own purposes.

The contributions of this paper are in the following:

- First proposing a novel program access authorization protocol to guarantee the security of smart meter parameters.
- Introducing the secret share scheme to authorize the

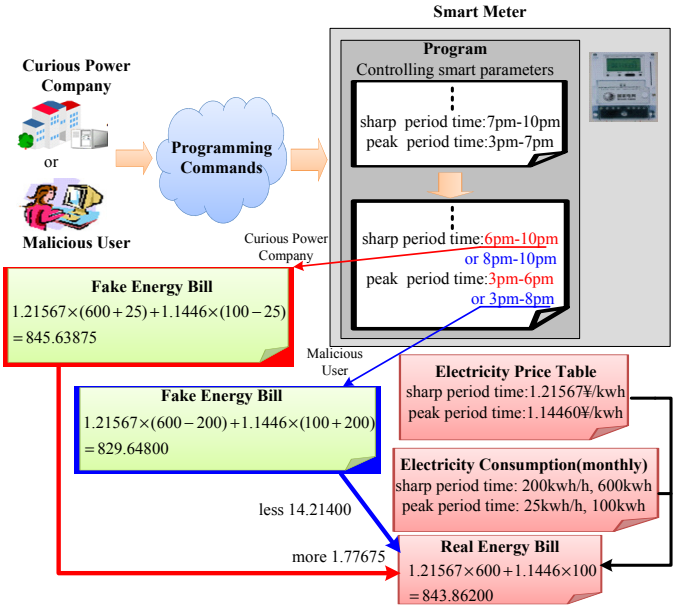


Figure 1: An example of unauthorized modifying smart meter parameters

modifying permissions of smart meter parameters.

- Quantitatively analyzing the security by introducing the binomial distribution metric.
- Experimental results show that the protocol is efficient.

The rest of this paper is organized as follows. Section 2 introduces the preliminary knowledge used in the protocol. Then we investigate the related work in Section 3. We describe the system model and security requirements in Section 4 and then present our proposed protocol in Section 5. Security analysis and experiment evaluation are shown in Sections 6 and 7 respectively. Finally, we conclude the paper.

2 Preliminary

2.1 Shamir's (t, n) Threshold Secret Sharing Scheme

(t, n) threshold key sharing scheme based on Lagrange interpolation formula was proposed in 1979 by A. Shamir [18], where a secret key can be divided into n shares, and each share is distributed to one participant, only the designated number of participants like t or more together can reconstruct the secret key [19].

In order to understand our protocol clearly, we give the definition of *share* used throughout the paper as follows:

Definition 1. *A share is the result value y by computing the following polynomial on inputting a known x.*

$$f(x) = (K + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}) \bmod Q$$

where $a_1, a_2, ..., K \in \mathcal{F}_Q$, Q is a large prime, $\mathcal{F}_Q$ is a finite domain on Q, K is the secret value.

From the definition above we know that n *share*s are $y_1, y_2, ..., y_n$ computed from known $x_1, x_2, ..., x_n$ respectively, and the polynomial $y = f(x)$ can be reconstructed from any t known pairs $(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), ..., (x_{it}, y_{it})$ of n pairs $(x_1, y_1), ..., (x_n, y_n)$.

2.2 Binomial Distribution

Binomial distribution is a probability distribution with discrete random variables, symbolized by $X \sim B(n, p)$, where X is the result of the randomized trial, n is the number of independent repeated trials, p is the occurrence probability of an event in one trial. The following is the mathematical definition of binomial distribution.

Definition 2. *Assuming an event A. The occurrence probability of A is p $(0 < p < 1)$ in one trial, thus the nonoccurrence probability of A in one trial is $q = 1 - p$. The probability that A occurs k times in n independent repeated trials is:*

$$P = (X = k) = C_n^k p^k q^{n-k}, k = 0, 1, ..., n \quad (1)$$

The probability that A occurs no more than k times in n independent repeated trials is:

$$F = (X = k) = P(X \leq k) = \sum_{j=0}^{k} C_n^j p^j q^{n-j} \quad (2)$$

3 Related Works

In the studies of smart metering security protection, most of them focus on smart meter data security protection. We briefly review these concerned works from two aspects: Smart meter data privacy protection and smart meter data security defense.

3.1 Data Privacy Protection

Smart meter data privacy protection aimed at achieving the power company's billing purposes and preserving users' privacy in the meantime.

The scheme in paper [5] preserved users' daily electricity usage pattern from a power operator using anonymous credentials while vast credentials need to be generated beforehand. The scheme in paper [6] using pseudo random identity requested power, which hid the smart meter's true identity and thus preserved users' privacy. The scheme in paper [8] assumed that each smart meter had two separate IDs, one of which attached to private information (i.e. HFID) was anonymous to preserve the users' privacy. In paper [24], the key distribution center without the smart meter's true identity performed billing computation and sent the total power consumption to the power company, which protected the real-time

power consumption from power companies and thus preserved users' privacy. The scheme in paper [11] used data aggregation which was performed at all smart meters to construct a aggregation tree, where smart meters as aggregation nodes routed metering data from the source meter to the power company and thus the power company only saw the final results. The schemes in papers [21] and [10] used rechargeable batteries to protect users power load information. Rechargeable batteries and the power company could individually or collectively supply electricity to smart appliances with the reasonable battery policy. Thus smart meter data could not directly reflect the smart appliances electricity usage information, which fuzzed users' electricity usage pattern. The scheme in paper [17] analyzed the trade-off relationship between information leakage and practicability using a stationary Gaussian Markov model of the electricity load. The scheme in paper [9] split the amount of electricity which was requested by users into random shares, one share for each user. Users submitted the mixed shares to the power company. Thus the power company only saw the final power consumption but could not retrieve the meaningful information about the real-time power consumption.

3.2 Data Security Defense

Smart meter data security defense aimed at protecting the smart meter data from physical attacks.

According to paper [15], smart meters could be equipped with seals, uncapping recordings, hardware programming switches or password authentication switches to protect smart meter data from being physically compromised. Both paper [25] and paper [13] suggested that the programming switch was used in conjunction with password authentication [22] to enhance defense capabilities. Paper [1] pointed out that using magnetic sensors or tilt sensors could check whether the authorized location of smart meters had been removed or physical tampered with.

Above all, there are almost no researches focusing on the security of smart meter parameters. The methods in smart meter data security defense also can be used to protect smart meter parameters from physical attacks. But smart meter parameters need to be protected from the aspect of software to against cyberattacks. This paper first proposes a secret share based program access authorization protocol to solve the problem. The protocol implements that users and power companies jointly control the modifying permissions on smart meter parameters.

4 System Model and Security Requirements

4.1 System Model

Figure 2 depicts the system architecture of the protocol where three types of participants exist: Smart meter, power companies and user.

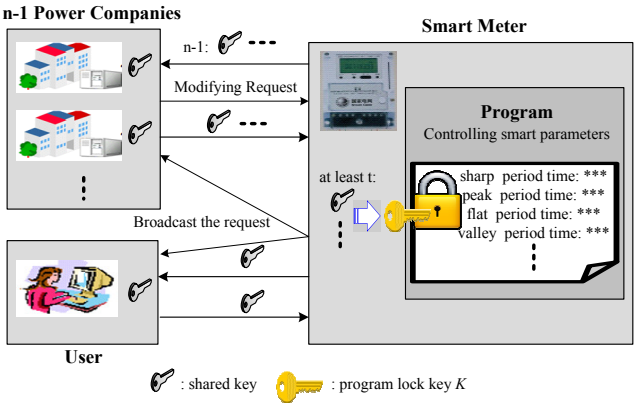


Figure 2: System architecture

- **Smart meter.** A smart meter locks the smart meter parameter modifying program by using the program lock key K and then divides the program lock key K into n shared keys. Each shared key is distributed to one participant. The smart meter asks for shared keys to recover the program lock key K when receiving a modifying request from one power company.

- **Power companies.** Power companies are semi-honest. They may modify smart meter parameters arbitrarily for their own purpose (e.g. increase users' energy bills though modifying smart meter parameters to earn the difference). There are $n-1$ power companies in the authorization protocol. Any one from these power companies can submit a modifying request to the smart meter. Other $n-2$ power companies submit shared keys to the smart meter according to the rationality of the reasons of modifying smart meter parameters.

- **User.** Users can modify smart meter parameters through modern devices, such as a new intelligent web-based electric meter concentrator according to paper [14], but they may arbitrarily modify for their own interests (e.g. decrease their own energy bills through modifying smart meter parameters to pay less than what they need to pay). In our protocol, the user is also a participant of the recovery of the program lock key K. The user submits his/her shared key to the smart meter according to the rationality of the reasons of modifying smart meter parameters.

It should be pointed out: A modifying request mainly includes programming commands and the reasons of modifying smart meter parameters; the total/sharp/peak/flat/valley period time varies with regions and seasons; the rationality of modifying reasons is determined by whether they confirm to the seasonal change or national policy.

4.2 Security Requirements

We aim at designing a program access authorization protocol to protect smart meter parameters from being modified arbitrarily. The security requirements are summarized as follows:

4.2.1 Request Message Authentication

Every modifying request message from one power company should be authenticated [3, 4] to confirm that it is from a valid entity. Thus an attacker cannot impersonate any valid power company to send out fake modifying request messages.

4.2.2 Shared Key Message Confidentiality

Shared keys distributed or recalled by the smart meter should be kept confidential to protect the security of the program lock key K [16]. Thus an attacker should not be able to get the shared keys to recover the program lock key K.

4.2.3 Anti-attacking Ability

The protocol should have anti-attacking abilities to protect the participants against outside attackers and thus protect the security of shared keys. An attacker should not be able to obtain the shared keys through attacking the participants.

5 The Protocol

5.1 The Notations in the Protocol

The notations used in the protocol are shown in Table 1.

Table 1: The notations in the protocol

Notation	Description
M	Smart meter
A/A_i	Power company/ the i^{th} power company
U	User who uses the smart meter
Program	A program which controls modifying smart meter parameters
K	A program lock key which is used to lock the program
Pub_R/Pri_R	Public key/private key pair of an entity R
$E_k(program)$	Encrypting the program under the program lock key K
$E_{Pub_R}(J)$	Encrypting the information J under the public key of an entity R
Sig_{Pri_R}	Signature of an entity R

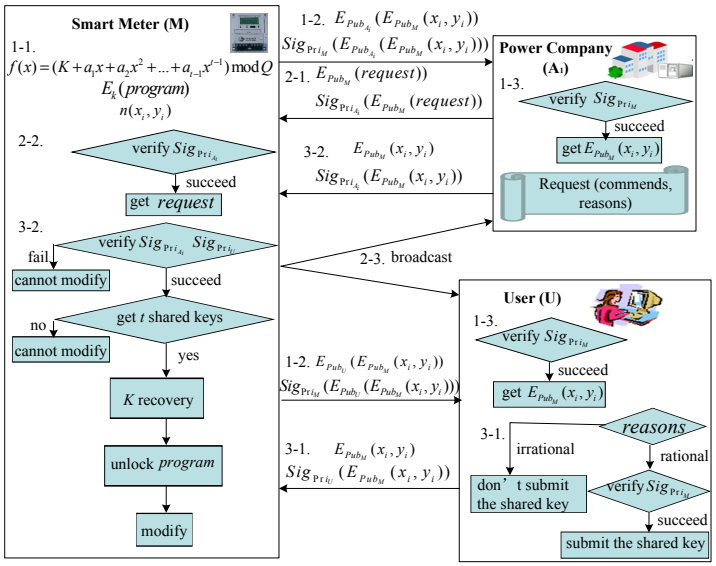


Figure 3: The protocol

5.2 The Protocol

Figure 3 shows the secret share based authorization protocol for smart metering. Actually there are $n-1$ power companies in our protocol. We only show one power company in Figure 3 for simplifying instructions. The following is the detailed processes of the protocol.

5.2.1 Generating And Distributing Shared Keys

Each protocol participant has its own private/public key and others' public keys (this process based on public key cryptography is not our focus, so we do not discuss it in detail). Below is the description of generating and distributing shared keys.

Step 1-1/2. A smart meter M randomly generates a $t-1$ power of polynomial, $f(x) = (K + a_1x + a_2x^2 + ... + a_{t-1}x^{t-1}) \bmod Q$ (Q is a prime number, $K < Q$). The smart meter uses the program lock key K to lock the smart meter parameter modifying program, $E_k(program)$. Then the smart meter selects n different non-zero elements (x_i, y_i) $(x_i \in Q, 1 \leq i \leq n, y_i = f(x_i))$ called secret keys. Anyone who has at least t $(t \leq n)$ secret keys can recover the program lock key K [23]. The smart meter encrypts these n secret keys with its public key, $E_Pub_M(x_i, y_i)$ called shared keys. Then the smart meter encrypts these n shared keys with the corresponding participants' public keys and sends them to the participants separately with its signature. At last, the smart meter destroys any information about the program lock key K, such as the shared keys $E_Pub_M(x_i, y_i)$, the polynomial $f(x)$, and the secret keys (x_i, y_i).

Step 1-3. Upon the power companies and user receive the massages, they first verify the signature of the smart meter. After succeeding, they get their own

shared keys $EPub_M(x_i, y_i)$ using their own private keys. But they can never know (x_i, y_i), the confidential information of the program lock key K, without the private key of the smart meter.

5.2.2 Modifying Request

If one power company (e.g. A_1) wants to modify smart meter parameters, he has to send a modifying request and his own shared key [7] to the smart meter. Below is the description of this phase (Figure 3).

Step 2-1. Power company A_1 sends a modifying request to the smart meter. The request is encrypted with the smart meter's public key and sent with the power company's signature.

Step 2-2. Upon the smart meter receives the request, it first verifies the power company's signature and then gets the request using its private key.

Step 2-3. Then the smart meter broadcasts the modifying request and it's signature to all participants asking for shared keys.

5.2.3 Recovering The Program Lock Key K

Step 3-1. Upon receiving the broadcast, other power companies and user first judge the rationality of the modifying reasons. If anyone considers the reasons are irrationality, he/she does not submit his/her own shared key to the smart meter. Otherwise, he/she verifies the smart meter's signature and sends his/her own shared key with his/her signature to the smart meter.

Step 3-2. Upon receiving the massages from the power companies and user, the smart meter first verifies their signatures respectively. After succeeding, the smart meter gets the shared keys. If getting less than t shared keys, the smart meter can not recover the program lock key K. This means the modifying request fails. Otherwise, the smart meter recovers the program lock key K and unlocks the program to modify the smart meter parameters according to the modifying request.

6 Security Analysis

In this section, we evaluate the protocol generally according to the security requirements summarized in Section 4.

6.1 Request Message Authentication

Before a power company sends a modifying request message to the smart meter, the power company has to sign the message using his private key. The private key is only known by the power company. Hence an attacker does not know how to produce the signature of the power company without the power company's private key. Thus an attacker could not able to pretend the power company to transmit the request message.

6.2 Shared Key Message Confidentiality

Shared keys (i.e. $EPub_M(x_i, y_i)$) are encrypted using the public key of the smart meter, which is significant from the conventional secret share scheme, no one can get the secret keys (i.e. (x_i, y_i)) except the smart meter. Thus, an attacker can not acquire the secret keys, the confidential information of the program lock key K, through eavesdropping or intercepting the shared key messages.

6.3 Anti-attacking Abilities Analysis

The program lock key K is used to lock the smart meter parameter modifying program. If getting the program lock key K, attackers can open the program to do something with smart meter parameters and then achieve their malicious purposes. The security of the program lock key K is crucial to the protocol. According to the targets which attackers attack to, we analyze the anti-attacking abilities from the following two aspects.

6.3.1 Targets Are Smart Meters

A smart meter randomly generates a polynomial and then gets the program lock key K and shared keys from this polynomial. After using the program lock key K to lock the smart meter parameter modifying program and distributing the shared keys to all participants, the smart meter destroys the program lock key K, the polynomial (i.e. $f(x) = (K + a_1x + a_2x^2 + ... + a_{t-1}x^{t-1}) \bmod Q$), the secret keys (i.e. (x_i, y_i)) and the shared keys (i.e. $EPub_M(x_i, y_i)$). This means that the smart meter does not store any information about the program lock key K. Thus, attackers can not get any information about the program lock key K through attacking [2] smart meters.

6.3.2 Targets Are Power Companies/Users

A power company/user receives a shared key from the smart meter and keeps it for future use (i.e. sends the shared key back to the smart meter for modifying smart meter parameters). The shared key (i.e. $EPub_M(x_i, y_i)$) is encrypted using the public key of the smart meter, no one except the smart meter can decrypt it. Thus, even though obtaining the shared key through attacking the power company/user, attackers can not get the confidential information (i.e. (x_i, y_i)) of the program lock key K.

6.4 Less Than t Participants Colluding Together Can Never Modify Smart Meter Parameters

From Section 5, we know that if a smart meter gets less than t shared keys, it can not recover the program lock key K to open up the smart meter parameter modifying

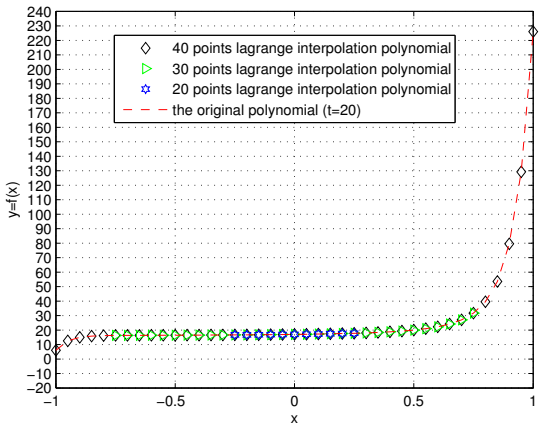


Figure 4: Success of recovering the program lock key K

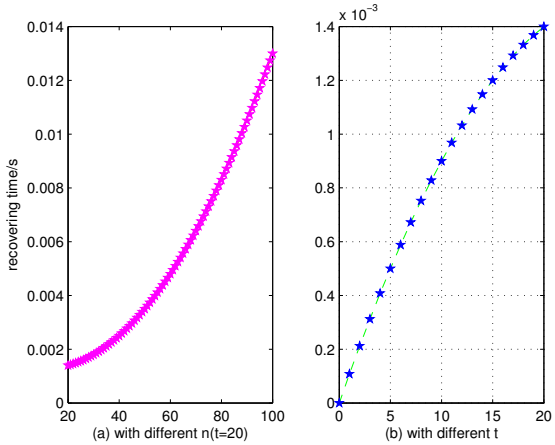


Figure 5: The time cost for the program lock key K recovering

program. So less than t malicious participants colluding together can not unlock the program to modify smart meter parameters. Any one power company/user cannot modify smart meter parameters arbitrarily through sending his/her only one shared key. Moreover, in real life, most power companies are trusted entities. So the probability of the smart meter parameters arbitrarily being modified is very small.

In addition, the program lock key K can not be derived out by participants. Since the shared keys are encrypted with the public key of the smart meter, the power companies and user can not get the confidential information of their own stored keys. So the program lock key K can not be deduced out by the participants colluding together through some algorithms (e.g. exhaustive algorithm), which is superior to the conventional secret share scheme.

7 Experiment Evaluation

7.1 The Determination of the Optimal t

We balance the time cost of the program lock key K recovering and the security of the program lock key K to determinate the optimal threshold t.

Figure 4 shows an 19th (i.e. $t = 20$) polynomial $y = f(x)$ is reconstructed by 20, 30 and 40 points from this polynomial respectively. From Figure 4, we can see that the curves of 20, 30 and 40 points lagrange interpolation polynomials overlap at zero, which proves that 20, 30 and 40 points can recover the program lock key K successfully. So we use the lagrange interpolation polynomial to test the time cost for the program lock key K recovering in MATLAB R2013a. The test results are shown in Figure 5.

From Figure 5(a), we can see that: When t ($t = 20$) is certain, the time cost increases as n increases, where n is the number of all participants in the protocol (i.e. one user and $n-1$ power companies). For example: When $n = 40$, the time cost for recovering is about 2.5 milliseconds; when $n = 100$, the time cost for recovering is about 13 milliseconds. So we assume that the smart meter choose t but not n shared keys to recover the program lock key K to reduce the recovering time cost.

From Figure 5(b), we can see that: When t is not certain, the time cost increases as t increases. Just take the time cost of the program lock key K recovering into consideration, the threshold t is as small as possible. But if consider the security of the program lock key K in the meantime, the threshold t is really not as small as possible.

Paper [12], which uses a precise number complementary judgment matrix sorting algorithm, established a decision analysis method for the selection of the $t \setminus n$ value in the threshold key sharing scheme. But this decision analysis method does not quantitatively analyze the security of Shamir's (t, n) threshold secret sharing scheme. We propose a method that uses binomial distribution, a simple mathematical algorithm, to quantitatively analyze the security of the program lock key K. The experiments we do are as follows.

Binomial distribution is symbolized by $B(n, p)$ mentioned in Section 2. Here we convert the symbol $B(n, p)$ into the analysis of the security of the program lock key K. We assume that n is the number of all participants in the protocol as before, and p is the probability of one secret key (i.e. (x_i, y_i)) leakage. Less than t secret keys leakage can not threat the security of the program lock key K, so the formula of the security degree (sd) of the program lock key K is as follows and the explication is shown in the following box. The experimental results are shown in Figure 6.

$$sd = P(X \leq t) - P(X = t)$$
$$= \sum_{j=0}^{t} C_n^j p^j (1-p)^{n-j} - C_n^t p^t (1-p)^{n-t} (t = 1, ..., n) \quad (3)$$

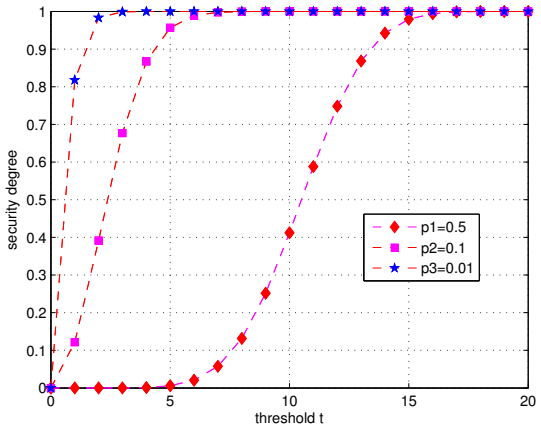


Figure 6: The security of the program lock key K

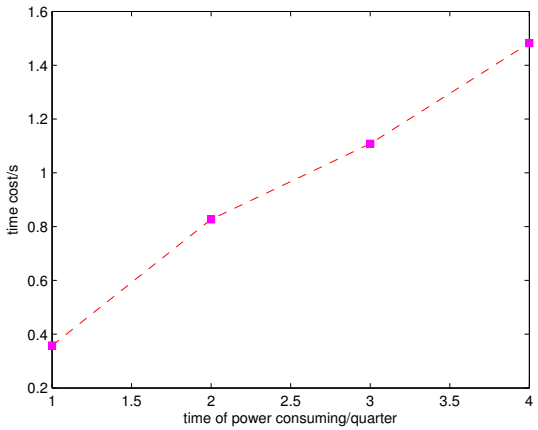


Figure 7: The time cost for encryption and decryption

The probability (degree) of security
= The probability of $(t-1)$ secret keys leakage
+ the probability of $(t-2)$ secret keys leakage
+ ... + the probability of one secret key leakage
= The probability of no more than t secret keys leakage - The probability of t secret keys leakage

Figure 6 shows the security of the program lock key K when $n = 20$. From Figure 6, we can see that: When n and p are certain, as t increases, the security degree is from zero to the maximum and then remains unchanged; when n is certain and p is very small, choosing small t can reach a high security degree. As shown in Figure 6, when $p = 0.5$, the security degree reaches the maximum (about 100%) when $t = 17$. So $t = 17$ is the optimal threshold when $n = 20$ and $p = 0.5$. From Figure 5(b), we can also get the time cost for recovering is about 1.3 milliseconds when $t = 17$. In addition, the secret key (i.e. (x_i, y_i)) is encrypted under the public key of the smart meter and no one except the smart meter can decrypt it, thus the probability of one secret key (i.e. p) leakage is quite small. So the protocol can have very high security degree with a small threshold (i.e. t).

7.2 The Time Cost for Encryption and Decryption

We deploy the environment with a ThinkPad Core 2 CPU E425 @1.90GHz PC, and choose RSA (1024 bits) as an the asymmetric encryption algorithm, coding in C.

The time cost for encryption and decryption in the fist phase of the protocol, namely the generating and distributing shared keys phase, is only about 94 milliseconds. Moreover, the generating and distributing shared keys phase is only carried out once in a smart meter unless the program lock key K leaks. Therefore, The time cost for encryption and decryption in the generating and distributing shared keys phase has little influence on the protocol efficiency.

In general, the second phase and the third phase of the protocol, namely the modifying request phase and the recovering the program lock key K phase, carry out once every three months (i.e. a quarter of a year) in China. We take encrypting and decrypting the longest information for example to test the time cost for encryption and decryption in these two phases during one year. The test results are shown in Figure 7. From Figure 7, we can see that the cost time for encryption and decryption is roughly linear with the quarter. The ratio of the cost time and the effective time is the average gradient (ag) of the curve in Figure 7, $ag = (ag_1 + ag_2)/2$, $ag_1 = (y_3 - y_1)/(x_3 - x_1)$, $ag_2 = (y_4 - y_2)/(x_4 - x_2)$, where x_i and y_i are the power consuming time (effective time) and the time cost for encryption and decryption respectively. This ratio is almost a constant value and so small, only about $5.26 \times 10^{-6}\%$. So the protocol has good efficiency.

8 Conclusions

Smart meter parameters, such as the total/sharp/peak/flat/valley period time, affect the authenticity of energy bills of users. Users or power companies can modify smart meter parameters through sending programming commands. However, they should not arbitrarily modify smart meter parameters for their own purposes. This paper proposes a secret share based program access authorization protocol for smart metering. The protocol realizes that one user and power companies jointly control the modifying permissions on smart meter parameters. Our future work is to construct the hierarchy based administrator domain [20] to enforce the program access authorization mechanism.

Acknowledgments

This work was supported by NSFC Grants (No. 61202020), Project of Shanghai Science and Technology Committee Grant (No. 15110500700) and CCF-Tencent Open Fund Grant (No. IAGR20150109, RAGR20150114).

References

[1] M. Balakrishnan, "Security strategy of smart meters," *China Electronic Market*, pp. 56–57, 2012.

[2] M. Bayat and M. R. Aref, "An attribute based key agreement protocol resilient to kci attack," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 10–20, 2015.

[3] C. C. Chang, J. H. Yang and Y. C. Wu, "An efficient and practical authenticated communication scheme for vehicular ad hoc networks," *International Journal of Network Security*, vol. 17, no. 6, pp. 702–707, 2015.

[4] Q. F. Cheng and C. M. Tang, "Cryptanalysis of an id-based authenticated dynamic group key agreement with optimal round," *International Journal of Network Security*, vol. 17, no. 6, pp. 678–682, 2015.

[5] J. C. L. Cheung, T. W. Chim, S. M. Yiu, C. K. Hui and V. O. K. Li, "Credential-based privacy-preserving power request scheme for smart grid network," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–5, Houston, 2011.

[6] T. W. Chim, S. M. Yiu, L. C. Hui and V. O. Li, "PASS: Privacy-preserving authentication scheme for smart grid network," in *Proceedings of the IEEE Smart Grid Communications Conference*, pp. 196–201, Brussels, 2011.

[7] X. D. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[8] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications*, pp. 238–243, Gaithersburg, MD, 2010.

[9] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proceedings of the 6th Workshop on Security and Trust Management (STM'10)*, pp. 226–238, Athens, 2010.

[10] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis and R. Capeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Proceedings of the IEEE International Conference on Smart Grid Communication*, pp. 232–237, Gaithersburg, MD, 2010.

[11] F. Li, B. Luo and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications*, pp. 327–332, Gaithersburg, MD, 2010.

[12] Q. D. Li and Y. H. Zhou, "Research and application based on a. shamir's (t, n) threshold secret sharing scheme," in *Proceedings of the IEEE International Conference on Computer Science & Education (ICCSE'12)*, pp. 14–17, Melbourne, 2012.

[13] X. F. Li, "Smart energy meter data security protection technology," *Urban Construction Theory Research*, 2013.

[14] J. H. Lin, "Icpdas issued a new intelligent web-based electric meter concentrator," *DIGITIMES*, 2014.

[15] W. Liu, "Smart energy meter data security protection technology," *Power Supply Technologies and Applications*, 2013.

[16] D. Manivannan and P. Neelamegam, "An efficient key management scheme in multi-tier and multi-cluster wireless sensor networks," *International Journal of Network Security*, vol. 17, no. 6, pp. 651–660, 2015.

[17] S. R. Rajagopalan, L. Sankar, S. Mohajer and H. V. Poo, "Smart meter privacy: A uility-privacy framework," in *Proceedings of the IEEE International Conference on Smart Grid Communication*, pp. 190–195, Brussels, 2011.

[18] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[19] X. X. Tian, C. F. Sha, X. L. Wang and A. Y. Zhou, "Privacy preserving query processing on secret share based data storage," in *Proceedings of International Conference on 16th Database Systems for Advanced Applications*, pp. 108–122, Hong Kong, China, 2011.

[20] X. X. Tian, X. L. Wang and A. Y. Zhou, "DSP re-encryption based access control enforcement management mechanism in DaaS," *International Journal of Network Security*, vol. 15, no. 1, pp. 28–41, 2013.

[21] D. Varodayan and A. Khisti, "Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1932–1935, Prague, 2011.

[22] Y. Wang and X. G. Peng, "Cryptanalysis of two efficient password-based authentication schemes using smart cards," *International Journal of Network Security*, vol. 17, no. 6, pp. 728–735, 2015.

[23] J. Q. Xie, "A practical key-sharing method," *Microcomputer Applications*, vol. 21, no. 6, 2005.

[24] C. M. Yu, C. Y. Chen, S. Y. Kuo and H. C. Chao, "Privacy-preserving power request in smart grid networks," *IEEE Systems Journal*, vol. 8, no. 2, pp. 441–449, 2014.

[25] S. Zhang and Y. F. Li, "Smart energy meter data security protection technology," *Power Demand Side Management*, vol. 12, no. 2, 2010.

Xiuxia Tian, Professor, School of Computer Science and Technology in Shanghai University of Electric Power. She received her master degree from Shanghai Jiaotong University in 2005 and doctoral degree from Fudan University in 2011 respectively. Now she is a visiting scholar of two years working with group of Secure Machine

Learning (SecML) at Computer Science Division, University of California, Berkeley, email: xxtian@fudan.edu.cn. She has published more than 30 papers and some papers publish in international conferences and journals such as DASFAA, ICWS and SCN. Her main research interests: Database security, privacy preserving, applied cryptography, machine learning.

Lisha Li Graduate, School of Electronics and Information Engineering in Shanghai University of Electric Power. Her research interests mainly focus on the security and privacy protection for the smart meter.

Jinguo Li, Assistant Professor, School of Computer Science and Technology in Shanghai University of Electric Power. He received his bachelor degree and doctoral degree from Hunan University in 2007 and 2014 respectively. Email: lijg@shiep.edu.cn. He has published more than 10 papers and some papers are published in international conferences and journals such as WASA and SCN. His main research interests: Applied cryptography, cloud computing, network security.

Hongjiao Li, Associate Professor, School of Computer Science and Technology in Shanghai University of Electric Power. She received her master degree from Huazhong University of Science and Technology in 2002 and doctoral degree from Shanghai Jiaotong University in 2008. Email: hjli@shiep.edu.cn. She has published more than 30 papers in international conferences and journals. Her main research interests: Trust Computing, cloud computing and bigdata security, operating system security.

Chunhua Gu, Professor, School of Computer Science and Technology in Shanghai University of Electric Power. He received his master degree and doctoral degree from East China University of Science and Technology (ECUST) during 1988-2007. He was a visiting scholar at Florida International University in 2002 and University of Wisconsin at Madison in 2006. His main research interests: Smart grid security, cloud based data management, security calculation for smart grid user.

Secure Authentication Protocol Based on Machine-metrics and RC4-EA Hashing

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, Nasr City, Egypt[1]
Faculty of Science, Al-Azhar University, Cairo, Egypt[2]
Faculty of Science (Girls), Al-Azhar University, Cairo, Egypt[3]
1 Al Mokhaym Al Daem, Cairo Governorate, Egypt
(Email: elngar_7@yahoo.co.uk)
(Received Aug. 28, 2015; revised and accepted Nov. 27 & Dec. 15, 2015)

Abstract

Most authentication schemes are using physical token such as smart cards to restrict services. Although these schemes have been widely deployed for various kinds of daily applications; there are severe challenges regarding their infrastructure requirements and security. This paper proposes an efficient authentication protocol based on user's machine-metrics. The proposed protocol uses the machine physical metrics to identify machines in the network, which provides the basic capability to prevent an unauthorized machine to access resources. Thus, machine-merics based authentication for machine can be looked as an analog of biometric-based authentication for human. The proposed protocol is employing the *RC4-EA Hashing* function to secure the collected machine-merics. Since it is satisfying the basic requirements of a cryptographic hash function. Therefore, the purpose of the proposed protocol is theft-proofing and guarding against attacks based on stolen or lost tokens. Also, it offers strong protection against several attacks such as credential compromising attacks.

Keywords: Authentication, machine-merics authentication, RC4-EA hash function, user authentication

1 Introduction

Internet has become the most convenient environment for education, business, and content management system (CMS) around the world [?]. Thus, internet security is an important issue to prevent the confidential information from being accessed by illegal users [?]. Remote user authentication plays the most principal service on the internet. It is a process of identifying an authorized user for a particular web service on the internet [?].

Smart card based authentication scheme is one of the most convenient and effective authentication mechanism. Which used to restrict access of the web service [?]. Although these schemes have been widely deployed for various kinds of daily applications, such as e-commerce, e-health; there are severe challenges regarding their infrastructure requirements [?], security, and privacy. Therefore, a failure of any of these security goals may render the whole system completely insecure and unpractical [?].

A common feature of these schemes is that; their security based on the *tamper-resistance* assumption about smart cards, i.e. they assume that the security parameters stored on the smart card cannot be extracted. However, recent research results have demonstrated that common commercial smart cards shall no longer be considered to be fully tamper proof. Which means, the secret information stored on the smart cards memory could be revealed by reverse engineering, power analysis [?], techniques or fault injection attacks. Thus, this obstacle has restricted the application of smart card based authentication schemes [?].

As a consequence, these schemes are susceptible to some types of attacks such as offline password guessing attack, Smart card loss attack , replay attack, user impersonation attack, and denial of service attack [?]. Since, attack techniques have over grown to compromise a user credentials; it would not be enough to secure a user's credentials, but also to secure a user's machine [?] .

This paper proposed, a machine-metrics authentication protocol for remote user authentication. In the proposed protocol, the machine-metrics are collected and then hashed using *RC4-EA Hashing* function $RC4-EA\ Hashing$ [?]. Which is lightweight, structurally different from the broken hash class, and can reuse existing RC4 algorithm [?]. Therefor, the idea behind using machine-metrics authentication is to ensure integrity and authenticity of user credentials with his machine-metrics [?]. So that, for an attacker to compromise a user account; different independent metrics have to be compromised first before gaining full access to the user

account [?].

The major goal of this paper is proposing a novel protocol to remote authentication depending on machine-metrics, instead of using the traditional smart card for remote user authentication. The proposed protocol is powerful, reliable, privacy-preserving and theft-proof. Hence, machine-merics are hashed using *RC4-EA Hashing* function $RC4-EA\ Hashing$ to guarantee high security and usability. Which leads, to overcome the drawback of the credential compromising attack. Since, the new way of handling the machine-metrics gives higher privacy protection for authentication systems.

The rest of this paper is organized as follows: Section **??** presents an overview of Preliminary, Machine-Metrics Authentication, *RC4-EA Hashing* Function $RC4-EA\ Hashing$. Section **??** introduces the proposed authentication protocol. Section **??** gives the implementation and security analysis. Finally, Section **??** contains the conclusion remarks.

2 An Overview

2.1 Preliminary

- **Authentication:** From the transcripts of server S, S can believe information $info$ is not modified. More specially, S can believe $info$ is indeed from a specific machine M.
- **Security of authentication protocol:** In the presence of attacker A, from the transcripts of the protocol Π the information $info$ is tampered to $info'$ by attacker A. Therefor, the probability that A can fool S to believe $info'$ is from the machine M without any change is negligible $negl$.
- **Negligible function:** A negligible function $negl$ is defined by [?]:

$$iff\ \forall c \in N\ \exists\ n_0 \in N, \text{ such that:}$$
$$\forall n \succcurlyeq n_0,\ negl(n) \prec n^{-c}.$$

- **Authentication protocol:** if for any attacker A, there exists a negligible function $negl$ satisfying Equation (**??**):

$$Pr[AthFool_{A,\Pi}(n) = 1] \preceq negl(n) \qquad (1)$$

2.2 User Authentication

Remote user authentication plays the most significant process to verify the authorized users of a web service on the Internet. Authors in [?] proposed "Multi-Channel User Authentication Protocol based on Encrypted Hidden OTP" . Where, the protocol proposed an efficient one time password (OTP) based authentication over a multi-channels architecture. Which, applying the RC4-EA encryption method to encrypt the plain-OTP to cipher-OTP [?]. Then, Quick Response Code (QR) code is used as a data container to hide this cipher-OTP. Also, the purpose of the protocol is integrating a web based application with mobile-based technology to communicate with the remote user over a multi-channels authentication architecture [?].

2.3 Machine-metrics Authentication

Authentication is the process of confirming the identity of a person, machine, or other entity, which requesting access under security constraints. This is done for the purpose of performing trusted communications between parties for computing and telecommunications protocols [?].

In authentication protocols, all the transmissions of the data from a user's machine to the server can be reveal to attacker through interception. From the viewpoint of security strength, most common authentication protocols fail to guarantee a fault-secure method for keeping the login information away from the public [?]. To enhance the security strength of the authentication protocol, machine-metrics based authentication protocol is proposed.

Machine-metrics are metrics collected about a remote machine for the purpose of identification. Where machine-metrics based authentication uses the unique metrics of a machine to verify its identity. The metrics used in a machine-metrics based authentication protocols are unique, universal and permanent. Such metrics are suitable for authentication purposes as they cannot be lost or change. Hence, it would be possible to uniquely distinguish between all machines on a network.

Figure **??** shows a machine authentication between machine to server, and user authentication between human to machine.

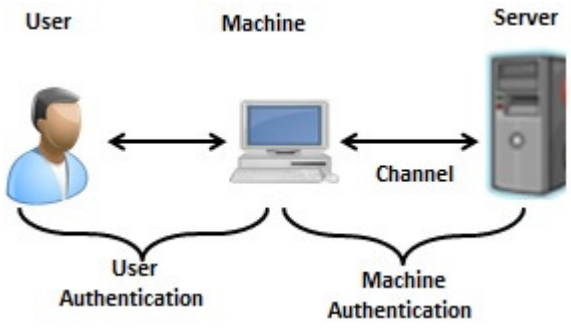


Figure 1: Machine authentication vs. user authentication

2.4 RC4-EA Hashing Function

Cryptography plays a significant procedure to prevent eavesdropping of sensitive information [?]. One of the fundamental components of many cryptographic protocols is a hash function [?].

Let $\{0,1\}^{\ell}$ denote the set of all messages of length strictly less than ℓ. A hash function is usually designed as follows; a compression function C :

$\{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{160}$ is designed. Then, given a message msg such that $|msg| < 2^{64}-1$, a pad is appended at the end of the message. Then, iterates the compression function C to get its output.

A cryptographic hash function has to be resistant against three main attacks [**?**]:

1) **Collision resistance:** For any msg_1, it should be 'hard' to find msg_2 where $msg_1 \neq msg_2$ and $H(msg_1) = H(msg_2)$.

2) **Preimage resistance:** For a given value $H(msg)$, it should be 'hard' to compute msg.

3) **Second preimage resistance:** Given msg_1 and $H(msg_2)$, it should be 'hard' to find msg_2 such that $msg_1 \neq msg_2$ and $H(msg_2) = H(msg_1)$.

RC4-EA Hashing function denoted as $(RC4 - EA\ Hashing)_\ell$, $16 \leq \ell \leq 64$ where $(RC4 - EA\ Hashing)_\ell : \{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{8\ell}$. $(RC4 - EA\ Hashing)_\ell$ function can be used to produce authenticator to authenticate the message msg. The $(RC4 - EA\ Hashing)_\ell$ function is describing as follows [**?**]:

1) **Padding rule:** the input message msg is padded by the padding bits generated by evolutionary algorithm (EA). a padding rule is applied to the message msg such that:

$$pad(msg) = bin_8(\ell)||msg||1||0^k||bin_{64}(|msg|)$$

where, $bin_{64}(|msg|)$ is the binary representation of number of bits of msg; and k is the least non-negative integer such that $8 + |msg| + 1 + k + 64 \equiv 0 \mod 512$. Then, $pad(msg) = msg_1||...||msg_t$ such that each $|msg_i| = 512$ bits, the maximum possible message length is $2^{64} - 1$.

2) **Classical iteration:** Let $msg_1||...||msg_t$ be the padded message. Let $(P_0, j_0) = (P^{IV}, 0)$ be an initial value. The iterations are followed as in Equation (**??**):

$$(P_0, j_0) \xrightarrow{msg_1} (P_1, j_1)...(P_{t-1}, j_{t-1}) \xrightarrow{msg_t} (P_t, j_t) = C^+(msg) \quad (2)$$

Where, $(P, j) \underrightarrow{B} (P^*, j^*)$ means that:

$$C((P, j), B) = (P^*, j^*)$$

Such that, $(B = B[0]||B[1]||...||B[63], B[i] = 8)$ and

$$C : Perm \times [N] \times \{0,1\}^{512} \longrightarrow [N]$$

3) **Post-processing:** Let the internal state after the classical iteration is (P_t, j_t) i.e., $OWT(C^+(msg)) = (P_t, j_t)$. Hence, the post processing is defined as follow:

- Computed $(P_{t+1}) = P_0 \circ P_t$ and $j_{t+1} = j_t$. Where, $\circ$ means the composition of the permutations.
- Define the final hash value $RC4 - EA\ Hashing_\ell(msg)$ by hash byte generation algorithm such that; $HBG_\ell(OWT(P_{t+1}, j_{t+1}))$.

The algorithms of the compression C , One-Way Transformation OWT and Hash Byte Generation HBG_ℓ functions are given in Algorithms **??**, **??** and **??** respectively [**?**]. Note all arithmetic is done modulo 256:

Algorithm 1 A compression function algorithm (C)

Imput: *Internal state (P, j), 64-byte message block B*
Output: *The updated internal state (P, j)*
1: **for** i = 0 to 255 **do**
2: $j = (j + P[i] + B[z(i)])$
3: $Swap(P[i], P[j])$
4: **end for**
5: Return (P, j)

Where, the function $z : [256] \rightarrow [64]$ is known as reordering, i.e. z is the mapping restricted on $[0, 63]$, $[64, 127]$, $[128, 191]$ and $[192, 255]$ are injective.

Algorithm 2 One-way transformation algorithm (OWT)

Imput: *Internal state (P, j)*
Output: *Updated internal state after padded*
1: $Temp_1 = P$
2: **for** i = 0 to 511 **do**
3: $j = (j + P[i])$
4: $Swap(P[i], P[j])$
5: **end for**
6: $Temp_2 = P$
7: $P = Temp_1 \circ Temp_2 \circ Temp_1$
8: Return (P, j)

Algorithm 3 A hash byte generation algorithm (HBG_ℓ)

Imput: *Internal state* (P, j)
Output: *The message digest*
1: **for** i = 0 to ℓ **do**
2: $j = (j + P[i])$
3: $Swap(P[i], P[j])$
4: **end for**
5: $H[i] = P[p[i] + p[j]]$
6: Return $H[i]$

2.5 Security of RC4-EA Hashing Function

Since the generation (hash value) of $RC4 - EA\ Hashing$ Function is close to uniform, it is impossible to find the

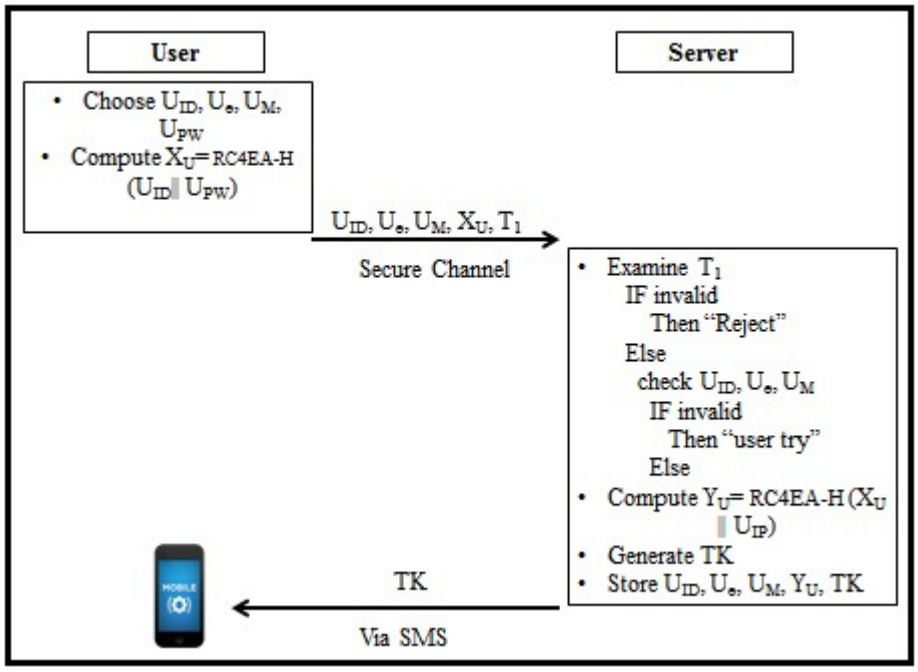


Figure 2: User enrollment phase

input through output and it is also computationally infeasible to find any two messages msg_1 and msg_2 such that $RC4{-}EA\ Hashing(msg_1) = RC4{-}EA\ Hashing(msg_2)$. Therefore, $RC4-EA\ Hashing$ function is one-way mapping and strongly collision free. Also, it is satisfies the basic requirements of a cryptographic hash function.

Therefore, the $RC4{-}EA\ Hashing$ Function is collision resistant, preimage resistant, and second preimage attack resistant. The efficiency of the $RC4-EA\ Hashing$ function is much better than widely used known hash functions and its structure is absolutely different from the broken hash function classes (e.g., SHA family). It is very secure out all possible generic attacks.

3 The Proposed Machine-metrics Authentication Protocol

The major aim of the proposed protocol is theft-proofing and guarding against attacks based on stolen or lost tokens. Also, it is defending the credential compromising attack A_{cc}; by introducing the machine-metrics. The proposed machine-metrics authentication protocol is enhancing a user authentication protocol proposed in [**?**].

The machine-metrics authentication protocol involves three parties: A server (S), a remote user (U) and Client Side Program CSP. The proposed protocol consists of three phases: User enrollment phase, machine-metrics enrollment phase, and machine-metrics authentication phase. The notations employed throughout this paper are shown in Table **??**.

Table 1: Notations

Notation	Description
U	Remote user
U_{ID}	User identity
U_{PW}	User password
U_{IP}	User IP address
U_M	User mobile
U_e	User electronic mail
S	The server
M	The machine
CSP	The client side program
$RC4-EA\ Hashing$	RC4-EA hashing function
V_{UHI}	Hashing for index the user
D_{HMC}	Hashing machine-metrics
$\|\|$	Concatenation
T	Time stamp
TK	Token
RNC	Random nonce code

3.1 User Enrollment Phase

In this phase, U enrollments at S in order to use a service. The enrollment process is shown in Figure **??** and have execute the following steps:

1) U chooses an identity U_{ID}, electronic mail U_e, mobile number U_M, and password U_{PW}. Then computes $X_U = RC4{-}EA\ Hashing\ (U_{ID}||U_{PW})$. Then sends

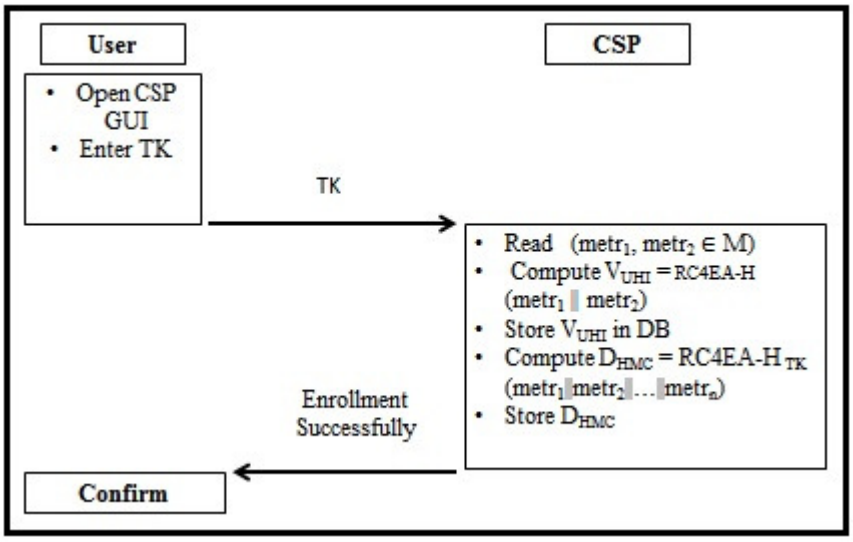


Figure 3: Machine-metrics enrollment phase

$\{U_{ID},\ U_e,\ U_M,\ X_U, T_1\}$ to S via a secure channel.

$$U \ \rightarrow \ S \ : \ \{U_{ID}, U_e, U_M, X_U, T_1\}$$

2) S examine the time stamp T_1. If it is invalid, then rejects it. Otherwise, checks whether U_{ID}, U_e, U_M is available for use. If it is, S computes $Y_U = RC4 - EA\ Hashing(X_U||U_{IP})$. Finally, S stores the values U_{ID}, U_e, U_M and Y_U in its database.

$$S \ \rightarrow \ DB \ : \ \{U_{ID}, U_e, U_M, Y_U\}$$

3) S generate random token TK, then sends TK to U via mobile channel.

$$S \ \rightarrow \ U \ : \ \{TK\} \tag{3}$$

4) Finally, S stores the values TK in its database.

$$S \ \rightarrow \ DB \ : \ \{TK\}$$

3.2 Machine-metrics Enrollment Phase

In this phase, the physical metrics of a machine are collected to be used as the identification of the machine. Suppose the physical metrics space is M which consists of n metrics; $M = \{metr_1, metr_2, ..., metr_n\}$. The client side program CSP will returns $metr_i \in M, (i = 1, 2, ..., n)$. The enrollment process is shown in Figure **??**. Then, U, S and CSP execute the following steps:

1) U received his TK from S via mobile channel.

2) U will enter his TK to CSP to enrollment his machine.

3) CSP read $metr_1$, $metr_2 \in M$. Then computes $V_{UHI} = RC4 - EA\ Hashing(metr_1||metr_2)$. Then stores the value V_{UHI} in a remote database DB.

$$CSP \ \rightarrow \ DB \ : \ \{V_{UHI}\}$$

4) CSP will use TK as a secret seed for RC4-EA Hashing, then computes:

$$D_{HMC} = RC4 - EA\ Hashing_{TK}(metr_1||metr_2||...||metr_n).$$

5) Finally, CSP stores the values D_{HMC} in a remote database DB.

$$CSP \rightarrow DB : \{D_{HMC}\}$$

3.3 Machine-metrics Authentication Phase

After U has a successful login. Now S wants to authenticate the machine upon client side program CSP. The machine-metrics authentication process is shown in Figure **??**. Then, U, S and CSP execute the following steps:

1) CSP read $metr_1$, $metr_2 \in M$. Then computes $V'_{UHI} = RC4EA - H(metr_1||metr_2)$.

2) CSP checks whether $V'_{UHI} == V_{UHI}$. If it is, then CSP will get the TK.

3) CSP computes:

$$D'_{HMC} = RC4{-}EA{-}Hashing_{TK}(metr_1||\ metr_2||...||metr_n)$$

using TK as a secret seed.

4) CSP checks whether $D'_{HMC} == D_{HMC}$. If it is, then CSP will generate random nonce code RNC

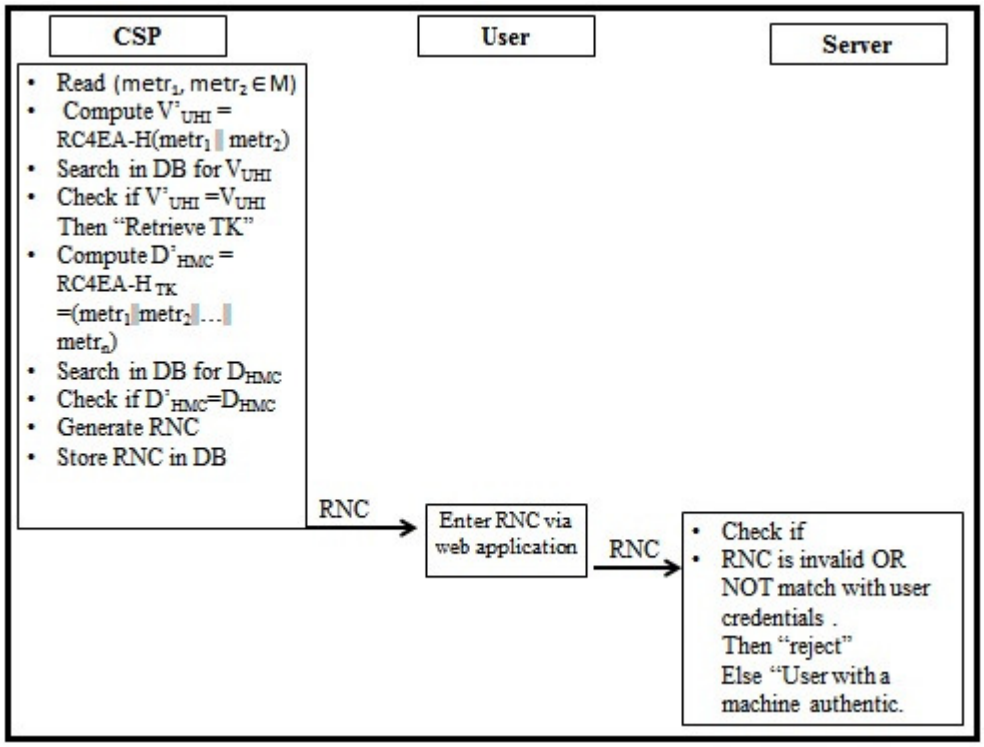


Figure 4: Machine-metrics authentication phase

to U with the $status = 1$ using *RNGCryptoService-Provider* which gives an unguessable crypto strength seed. Hence, it gives the random object with a different crypto strength number each time. Which mean is that, it will go on to return a different random number for each call. Then CSP stores the values RNC in a remote database DB.

$$CSP \rightarrow DB : \{RNC\}$$

5) U sent RNC to S via web application.

6) Finally, S checks whether RNC is invalid or not match with user credentials at the DB then, "request is rejected". Otherwise, user's machine is authentic and convert RNC status to 0.

4 Implementation and Security Analyses

The proposed machine-metrics authentication protocol is adopting the RC4-EA Hashing function $RC4 - EA\ Hashing$ to hash the machine physical metrics. The machine-metrics takes the responsibility for achieving mutual authentication between the M and S.

The performance of the proposed authentication protocol is tested using server 32 core AMD opteron processor 6376 with 32 GB of RAM and 4 RAID 1s, laptop (Intel i5, 1.80 GHz processor, 2 GB RAM) and simple mobile phone. The experiments have been implemented using PHP-MySql and C-sharp language environments.

4.1 Implementation

The proposed machine-metrics Authentication protocol is implementing the RC4-EA Hashing function $RC4 - EA\ Hashing$ to hash the machine physical metrics, which are collected via the client side program CSP like {Total Machine Memory (TMM), Procesor Id (PId), Name of CPU ($NCPU$), MotherBoard Id (MId), Hard Desk Id ($HDId$) }as shown in Table **??**. Thus, it helps in mitigating the credential compromising attack A_{cc}. Whenever a user U wishes to login the website, first step is to enter U_{ID} and U_{PW} for remote User authentication. The second step is that; a machine physical metrics will be collected via client side program CSP. Then, the CSP will hash the machine physical metrics using $RC4 - EA\ Hashing$. The third step is that; CSP will generate RNC to authentic the machine as shown in Tables **??**, **??**, **??**. Thus, the proposed machine-metric authentication protocol is integrating a web based application with desktop based application to make it more secure than the general authentication protocols.

Table 2: CSP collected machine physical metrics table

TMM	PId	$NCPU$	MId	$HDId$
21...48	BF...652	In.(R)C.i3-M330@2.GHz	.4X...4876.	20...058
32...72	BF...655	In.(R)C.i3-M380@2.GHz	.4C...00RW.	20...436
85...92	BF...6A7	In.(R)C.i7-2MCPU@2.GHz	.PC...A0UG.	W7...5YK

Table 3: User register table to main website

U.N	**Password**	**Email**	**Mobil No.**	**Token** TK.
Jack	895*/66!	Jack@egywow.com	968935810	K8*roMS1
Henary	P**2334	Henary@egywow.com	968925612	D4A/gE7S
Bill	Ad2*!98	Bill@egywow.com	966954523	1B6loP3S

Table 4: Hashing machine metrics table via $RC4 - EA\ Hashing$

$U.N$	TK	V_{UHI}	D_{HMC}
Jack	K8*roMS1	600eaw73b...	MDWKcEMZ3...
Jack	D4A/gE7S	ds734be484...	PXmnnMCa4Rp...
Jack	1B6loP3S	83d91a2d58...	3i18E1aaZby...

Table 5: Machine authentication code table

V_{UHI}	TK	D_{HMC}	RNC
600eaw73b...	K8*roMS1	MDWKcMEMZ3...	Ez8U89w91
600eaw73b...	K8*roMS1	MDWKcMEMZ3...	5Vr2uo5XD
600eaw73b...	K8*roMS1	MDWKcMEMZ3...	x99ICN41C

4.2 Security Analyses

The security of the proposed protocol is analyzed under the possibilities of the types of attacks listed below:

1) **Prevent Man-in-the-middle attack:** In this type of attack, the attacker listens to the communication channel between S and U. In the proposed protocol, the attacker may intercept the mobile communication messages, but he will never be able to compute the D_{HMC}. Since, it is based on $RC4-EA\ Hashing$. So attacker should know the hash function and use same user's machine physical metrics. Hence, the proposed protocol is secure against man-in-the-middle attacks.

2) **Prevent phishing attack via the web:** This attack aims to steal sensitive information. In the proposed protocol, if the attacker knows U_{ID} and can get the U_{PW} from the server by replacing the actual web page with a similar one, it would be difficult to get the token TK because it send over mobile channel as in Equations (**??**). This, the proposed protocol is secure against the phishing attacks.

3) **Prevent credential compromising attack:** Denoted as A_{cc}, this attack aims to hacked, modified, exposed, or cloned softwares or hardwares for a machine identification. In the proposed protocol, as the hash function $RC4 - EA\ Hashing$ is secure, attacker cannot compute D_{HMC}. It can be looked as a computed credential of a machine to guarantee the authentication security. That is, $Pr[AthFool_{A_{cc},\Pi}(n) = 1] \preceq negl(n)$. Therefor, the proposed protocol is secure against credential compromising attacks.

4) **Prevent impersonation attack:** In this type of attack, the malicious user forges the security parameters from the authentication protocol and tries to impersonate as a legitimate user. In our protocol, the malicious user has to guess the parameters Y_U, V_{UHI} and TK which is used in the calculation of D_{HMC} for generating a valid login request. The RC4-EA hashing function $RC4 - EA\ Hashing$ is impossible to solve in real polynomial time, thus the$RC4 - EA\ Hashing$ parameter cannot be forged. Therefore, the malicious user will fail to launch an impersonation attack on this proposed protocol.

5 Conclusions

The major contribution of this paper, is proposing machine-metrics authentication protocol. The proposed protocol enhances the security of a remote user login; by using the physical metrics of a machine. Also, the proposed protocol is adopting the *RC4-EA Hashing* function to secure these machine metrics. Therefore, the data can not be easily retrievable without adequate authorization. The purpose of this paper is to integrate a web based application with desktop based application to make the proposed protocol more secure than the general authentication schemes. Thus, the proposed authentication protocol is more convenient, because the burden of carrying a separate hardware tokens are removed. Moreover, this protocol helps to overcome many challenging attacks such as phishing attacks and credential compromising attacks.

References

[1] A. Aboshosha, K. A. ElDahshan, E. K. Elsayed and A. A. Elngar, "Multi-channel user authentication protocol based on encrypted hidden OTP," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 13, no. 6, pp. 14–19, 2015.

[2] A. Aboshosha, K. A. ElDahshan, E. K. Elsayed and A. A. Elngar, "EA based dynamic key generation in RC4 ciphering applied to CMS," *International Journal of Network Security (IJNS)*, vol. 17, no. 4, pp. 405–412, 2015.

[3] M. Bellare and P. Rogaway, "Collision-resistant hashing: towards making UOWHFs practical," in

Advances in Cryptology (Crypto'97), LNCS 1294, pp. 470–484, Springer-Verlag, 1997.

[4] M. Bellare and T. Kohno, "Hash function balance and its impact on birthday attacks," in *Advances in Cryptology (Eurocrypt'04)*, LNCS 3027, pp. 401–418, Springer-Verlag, 2004.

[5] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Advances in Cryptology (Crypto'92)*, LNCS 740, pp. 390–420, Springer-Verlag, 1992.

[6] D. Chang, K. C. Gupta and M. Nandi, "RC4-Hash: A new hash function based on RC4," in *7th International Conference on Cryptology in India*, vol. 4329, pp. 80–94, Springer-Verlag, 2006,

[7] D. Chatterjee, J. Nath, S. Mondal, S. Dasgupta and A. Nath, "Advanced symmetric key cryptography using extended MSA method: DJSSA symmetric key algorithm," *Jounal of Computing*, vol. 3, no. 2, pp. 66–71, 2011.

[8] M. A. Fairuz and K. Renaud, "Multi-channel, multi-level authentication for more secure ebanking," In ISSA, 2010.

[9] 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.

[10] A. Hiltgen, T. Kramp, T. Weigold, "Secure internet banking authentication," *IEEE Transactions on Security and Privacy*, vol. 4, no. 2, pp. 21–29, 2006.

[11] S. Kalra, S. Sood, "Advanced remote user authentication protocol for multi-server architecture based on ECC," *journal of information security and applications*, vol. 18, pp. 98–107, 2013.

[12] M. Kumar, "An enhanced remote user authentication scheme with smart card," *International Journal of Network Security*, vol. 10, no. 3, pp. 175–184, 2010.

[13] H. Le, C. Chang and Y. Chou, "A novel untraceable authentication scheme for mobile roaming in GLOMONET," *International Journal of Network Security*, vol. 17, no. 4, pp. 395–404, 2015.

[14] J. Malik, D. Girdhar, R. Dahiya and G. Sainarayanan, "Multifactor authentication using a QR code and a one-time password," *Journal of Information Processing Systems*, vol. 10, no. 3, pp. 483–490, 2014.

[15] C. Ma, D. Wang and S. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2012.

[16] P. E. S. N. K. Prasasd, A. S. N. Chakravarthy and B. D. C. N. Prasad, "Performance evaluation of password authentication using associative neural memory models," *International Journal of Advanced Information Technology (IJAIT)*, vol. 2, no. 1, pp. 75–85, 2012.

[17] S. K. Sood, "Secure dynamic identity-based authentication scheme using smart cards," *Information Security Journal: A Global Perspective*, vol. 20, no. 2, pp. 67–77, 2011.

[18] X. Sun, S. Men, C. Zhao and Z. Zhou, "A security authentication scheme in machine-to-machine home network service," *Security and Communication Networks*, vol. 10, no. 16, pp. 2678–2686, 2012.

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 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 Universitde 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 Tlommunications 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 Center (IDSC); and he was a senior advisor to the Ministry of Education and Deputy Director of the National Technology Development Center. 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 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 is assist. Prof. Computer science, Al-azhar university, Master of computer science, Cairo University 1999, Bachelor of Science, mathematics and computer science Department, Cairo University 1994. I Published thirty four papers until 2015 in data mining, Ontology engineering, e-learning and software engineering. I also published two books in Formal methods and event B on Amazon database. I am a member of Egyptian mathematical society and Intelligent computer and information systems society. Finally, I'm a certified trainer in AQATC Alazhar Quality Assurance and Training Center.

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).

Octopus: An Edge-Fog Mutual Authentication Scheme

Maged Hamada Ibrahim
(Corresponding author: Maged Hamada Ibrahim)
Department of Electronics, Communications and Computers Engineering, Helwan University
1, Sherif St., Helwan, P.O. Box 11792, Cairo, Egypt
(Email: mhii72@gmail.com)
(Received Sept. 1, 2015; revised and accepted Dec. 7 & Dec. 15, 2015)

Abstract

Authentication is an important and challenging issue for the security of Fog computing since, services are offered to massive-scale end users (Fog users or Edge) by front Fog servers (or Fog nodes). In this paper, we propose a secure and efficient mutual authentication scheme for the Edge-Fog-Cloud network architecture, to mutually authenticate Fog users at the Edge of the network, with the Fog servers at the Fog layer. Our scheme requires a user – roaming randomly in the network – to hold only one long-lived master secret key (with long enough bit-length) allowing him to communicate with any of the Fog servers in the network, in a fully authenticated way. The Fog users are able to mutually authenticate with new Fog servers joining the network, without the need to re-register and without any extra overheads. Moreover, the servers in the Fog are required to store only one secret key for each Fog user. On the other hand, the Fog users are totally unrelated to any public-key infrastructure. The scheme requires the Fog user to perform very few hash invocations and symmetric encryptions/decryptions. Therefore, the scheme is suitable to be efficiently implemented on the Fog user's smart card/device.

Keywords: Cloud computing, Edge layer, Fog layer, mutual authentication, rogue nodes, smart cards

1 Introduction

Through the last decade, Cloud computing has provided many opportunities for enterprises, by offering their customers a range of computing services. "Pay-as-you-go" Cloud computing model becomes an efficient alternative to owning and managing private data centers for customers facing web applications and batch processing [1, 33]. Cloud computing frees the enterprises and their end users from the specification of many details, such as storage resources, computation limitation and network communication cost. However, this felicity becomes a problem for latency-sensitive applications, which require nodes in the vicinity to meet their delay requirements [3]. When techniques and devices of IoT are getting more involved in people's life, with millions of such devices acquiring services, current Cloud computing architecture can hardly satisfy their requirements of mobility support, location awareness and low latency [33].

The Fog is a layer intermediate between the end users (Edge of the Network) and the Cloud, to bring far in cyberspace Cloud services to close proximity to the Edge and on a wider range. In Fog computing, services can be hosted at end devices such as, access points as illustrated in Figure 1. The infrastructure of this new multi-layered distributed computing allows applications to run as close as possible to sensed actionable and massive data coming out of people, processes and thing. Both Cloud and Fog provide data, computation, storage and application services to the Edge. However, Fog can be distinguished from Cloud by its proximity to end-users, the dense geographical distribution and its support for mobility. There exists wide range of applications for the Fog services, some of which are described next:

- *Malls*: Assuming that a number of Fog servers are deployed inside a multi-floor shopping center, which collectively form an integrated localized information system. The Fog servers at different floors can pre-cache floor-related contents, such as the layout, offers, ads, goods prices, etc. of stores on a particular floor. The Fog servers can deliver engaged services including indoor navigation, ads distribution and feedback collections to mobile users through, for example, WiFi.

- *Airports/Park zones*: The Fog computing system can be deployed in the parkland/zones to provide localized travel services. For instance, Fog servers can be deployed at the entrance and other important locations of the park. The Fog server at the entrance can pre-cache information including park map, park free slots, travel guide and local accommodations; other

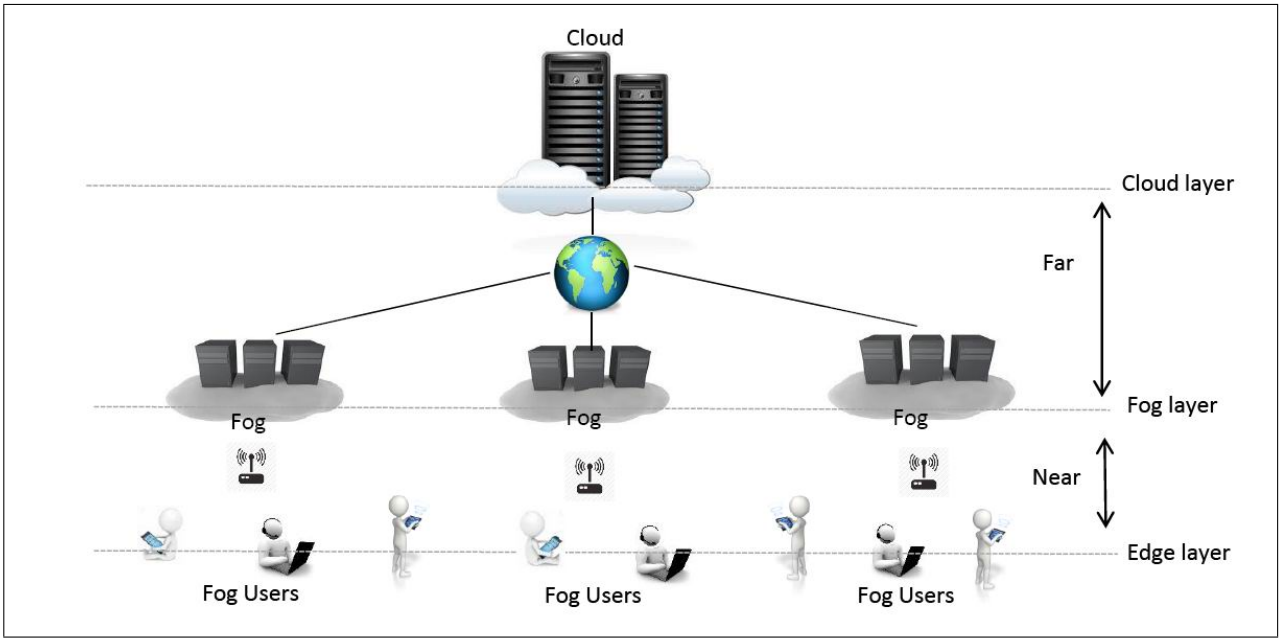


Figure 1: Edge-Fog-Cloud Architecture

Fog servers at different locations inside the park can be incorporated with sensor networks for environment monitoring and provide navigation to travelers. By connecting the Fog servers to the park administration office and Cloud, the Fog servers can be used as an information gateway to send timely alerts, notifications and information to travelers. Such services apply also to airports such as flight dates, delays, airport shops products and offers, medical services, etc.

- *Smart grid*: Energy load balancing applications may run on network Edge devices, such as smart meters and micro-grids [39]. Based on energy demand, availability and the lowest price, these devices automatically switch to alternative energies like solar and wind. Fog collectors at the Edge process the data generated by grid sensors and devices, and issue control commands to the actuators [3]. They also filter the data to be consumed locally, and send the rest to the higher tiers for visualization, real-time reports and transactional analytics. Fog supports ephemeral storage at the lowest tier to semi-permanent storage at the highest tier. Global coverage is provided by the Cloud with business intelligence analytics.

- *Transportation*: A Fog server can be deployed inside the bus and provides onboard video streaming, gaming and social networking services to travelers using WiFi. The on-board Fog server connects to the Cloud through cellular networks to refresh the pre-catched contents and update application services. Using its computing facility, the Fog server can also collect and process user's data, such as number of travelers and their feedbacks, and reports to Cloud [3, 33].

In Cloud computing deployment, data centers are usually owned by Cloud service providers. However, Fog service providers can be different parties, due to different deployment choices: Internet service providers or wireless carriers (e.g. GSM), who have control of home gateways or cellular base stations, may build Fog with their existing infrastructures. Cloud service providers, who want to expand their Cloud services to the Edge of the network, may also build Fog infrastructures. End users, who own a local private Cloud and want to reduce the cost of ownership, would like to turn the local private Cloud into Fog and lease spare resources on the local private Cloud. This flexibility complicates the trust situation of Fog and makes it different from other network architectures.

As a consequence of the absence of authentication services, a rogue Fog node/server would be a Fog device or Fog instance that pretends to be legitimate and masquerade Fog nodes for Edge users to connect to it. For example, in an insider attack, a Fog administrator may be authorized to manage Fog instances, but may instantiate a rogue Fog instance rather than a legitimate one. The discussion in [33] has demonstrated the feasibility of man-in-the-middle attack in Fog computing, before which the gateway should be either compromised or replaced by a Fake one. Once connected, the adversary can manipulate the incoming and outgoing requests from end users or Cloud, collect or tamper user data stealthily, and easily launch further attacks. The existence of fake Fog node or server is a serious threat to user data security and privacy.

2 Related Work

Security and privacy issues were not studied to directly hit the requirements of Fog computing. Some studies were in

the context of smart grids [38] and machine-to-machine communications [23]. There are security solutions for Cloud computing. However, they are not suitable for Fog computing because Fog devices work at the Edge of networks on a larger and wider scale. The environment of Fog devices is faced with many threats which do not exist in well managed Cloud. We next discuss the contributions we found closely related to Fog computing.

Reputation based trust model [16] has been applied successfully in e-Commerce, peer-to-peer, user reviews and online social networks. The work in [7] proposed a robust reputation system for resource selection in peer-to-peer networks using a distributed polling algorithm to assess the reliability of a resource before downloading. In designing a Fog computing still some problems are not solved; *How to achieve persistent, unique, and distinct identity? How to treat intentional and accidental misbehavior? How to conduct punishment and redemption of reputation?* There are also trusting models based on special hardware such as Secure Element (SE), Trusted Execution Environment (TEE), or Trusted Platform Module (TPM), which can provide trust utility in Fog computing applications.

Authentication is an important issue for the security of Fog computing since services are offered to massive-scale end users by front Fog nodes/servers. In [33] the authors have considered the main security issue of Fog computing as the authentication at different levels of Fog nodes. *Traditional PKI-based authentication is not efficient and has poor scalability for Fog users at the Edge of the network.*

The work in [2] proposed a cheap, secure and user-friendly solution to the authentication problem in local ad-hoc wireless network, relying on a physical contact for pre-authentication in a location-limited channel. Similarly, NFC can also be used to simplify the authentication procedure in the case of Cloudlet [4]. As the emergence of biometric authentication in mobile computing and Cloud computing, such as fingerprint authentication, face authentication, touch-based or keystroke-based authentication. However, such techniques take relatively long execution time and their security level is always constrained by time complexity, specially when high security level is needed.

Intrusion detection techniques can also be applied in Fog computing [24]. Intrusion in smart grids can be detected using either a signature-based method in which the patterns of behavior are observed and checked against an already existing database of possible misbehaviors. Intrusion can also be captured by using an anomaly-based method in which an observed behavior is compared with expected behavior to check if there is a deviation. The work in [35] develops an algorithm that monitors power flow results and detects anomalies in the input values that could have been modified by attacks. The algorithm detects intrusion by using principal component analysis to separate power flow variability into regular and irregular subspaces.

Password-based authentication techniques [17, 18, 22, 34] have many applications in the Cloud, however, they are not a good idea when it comes to Fog computing due to several reasons: (i) Passwords are characterized by their low entropy, and in order to amplify this entropy to establish session keys, extensive modular arithmetic computations are needed. (ii) Fog users at the Edge of the network communicate with many Fog servers in different Fogs. It is inadequate to keep a password with each server. Moreover, it is not a good idea to keep one common password for all servers. Also the Fog users may communicate in the future with newly joined servers that they never contacted before. (iii) Passwords in general are a weak link in Cloud computing [19] due to its vulnerability to off-line dictionary attacks[1].

Another closely research area is the Secure Wireless Roaming of mobile nodes [9, 36, 37, 41]. However, these protocols are realized by the session keys distributed using public key cryptosystems. The work in [28] presented the lightweight secure structure SPINS and the broadcast authentication protocol μ-TESLA. The μ-TESLA used a reverse hash chain to replace the public-key cryptosystem heavy algorithms. Other protocols are found in [10, 11, 20, 29]. Recently, the work in [40] exploited the advantages of Cloud-assisted BSNs based on MWN model, and designed an efficient, secure and composable protocol for the mobile nodes roaming randomly in the networks, still the protocol requires public key encryption. These protocols are not lightweight enough to be adequate for the massive scalability of Fog computing and the dynamic structure of the network.

Up to our knowledge and until the time this paper was written, the scheme in this paper is the first to directly target mutual authentication in the Edge-Fog layer of the Edge-Fog-Cloud architecture.

3 Motivations and Contributions

In this section, we discuss our motivations and the contribution of our work.

3.1 Motivations

Authentication is an important issue for the security of Fog computing since, services are offered to massive-scale end users by front Fog nodes. Fog users usually have smart devices that are limited in resources and hence, cannot perform extensive traditional digital signatures and public key encryptions. In addition, PKI is impractical to be implemented at the massive scale of the Edge. Password-based authentication strategies are not suitable for communication with large number of servers. Biometric based solutions also have the problem of very long execution time and their security level is always constrained by time complexity, specially when high security level

[1] http://searchcloudsecurity.techtarget.com/tip/password-basedauthentication- a-weak-link-in-cloud-authentication

is needed. Also authentication techniques using Diffie-Hellman key exchange [8], based on the DH problem, use extensive modulo computations which are slow and not suitable for smart devices/cards.

3.2 Our Contribution

We propose an efficient and secure Edge-Fog mutual authentication scheme, to allow any Fog user and any Fog server to mutually authenticate each other, without relying on any PKI. The Fog user is required to store only one long-lived master secret key, and this key allows him to roam randomly in the network and mutually authenticate with any Fog server in any Fog under the authority of a particular CSP. Also, the Fog user must be able to authenticate with new servers joining the Fog without the need to re-register and without any extra overheads. Our scheme uses elementary cryptographic tools (hash functions and symmetric encryption). The computations required by the Fog user are only few hash invocations and symmetric encryptions/decryptions. This makes our scheme very efficient for implementation on smart cards and devices with very limited resources, such as, sensor nodes and smart phones.

4 Model and Assumptions

In this section we describe the network model and assumptions of our scheme.

4.1 Network Model

The Cloud-based Internet is extended by introducing an intermediate layer between Edge users' (Fog users') devices and Cloud, aiming at the smooth, low-latency service delivery from the Cloud to Fog users. This accordingly leads to a three hierarchy Edge-Fog-Cloud architecture. Given a Cloud service provider (CSP), among his Cloud servers in the Cloud layer, there is a special server called the Registration Authority RA of the Cloud, which is responsible for registering Cloud users to the Cloud, as well as Fog users to the Fogs managed by this particular Cloud. Therefore, as shown in Figure 2, under the authority of the RA, there are several locations where for each location (or a Fog F), there is a set of Fog servers/nodes, $\mathcal{FS} = \{FS_1, ..., FS_n\}$. $\mathcal{FS}$ directly communicate with the Fog users, $\mathcal{FU} = \{FU_1, FU_2, ...\}$, in its location through single-hop wireless connections using the off-the-shelf wireless interfaces, such as WiFi, Zigbee or even Bluetooth. With the on-board compute facility and pre-cached contents, they can independently provide pre-defined service applications to mobile users without assistances from Cloud or Internet. On the other hand, the Fog servers, $\mathcal{FS}$, of a Fog F can be connected to the Cloud so as to leverage the rich functions and application tools of the Cloud.

Dynamic join and leave of Fog Servers. Unlike Cloud servers, the Fog servers/nodes are dynamic in joining and leaving the Fogs. Servers in different places (e.g. Shops, Groceries, Parklands, Bus stops, etc.) may be added to (removed from) a Fog at any time and this may happen frequently. Still services provided by these newly added servers must be available to the registered users. This property of Fog servers must be dealt with in an efficient way from the Fog users' perspective. We argue that this dynamic change of the Fogs must be transparent to the Fog users. I.e. Fog users must still be able to mutually authenticate themselves to the newly joined Fog servers, without the need to re-register any parameters, and without any increase in the complexity at the Fog users' side.

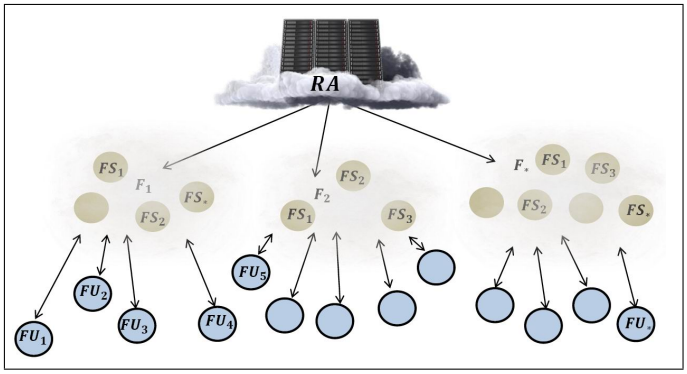


Figure 2: The network model

4.2 Assumptions

We assume that the registration authority RA communicates with all managed Fog servers through private and authenticated channels, that could be realized by establishing a public key infrastructure (PKI). In fact, we assume a PKI as a folklore realization of such channels, between the RA and her servers since we focus on the authentication in the Edge-Fog layer. There are many other ways to realize such channels, e.g. if the RA shares a master secret key with each of her servers, the private and authenticated channels are realized without a PKI. Also, PKI may be avoided if the CSP installs her public key pk_{RA} on the servers and the server's private key sk_{FS} on each server FS. The Fog servers in our protocol are not needed to communicate with each other by any means, they communicate only with the RA and with the Edge users when requested. The Fog users at the Edge of the network are completely unrelated to the established PKI. We assume that the RA as a service provider is trusted, however, none of the servers in the Fog are assumed trusted, they are vulnerable to corruption by a corruptive adversary.

5 Our Proposed Scheme

The notations used in our protocol is given in Table 1. Let (E, D) be the encryption/decryption function of a

Table 1: Notations used in our scheme

Notation	Meaning
RA	Registration Authority
FU	Fog User
FS	Fog Server/Node
F	Fog/location/zone/area
ID_{FU}	Identity of Fog user
ID_{FS}	Identity of Fog server
ID_F	Fog/zone/area identity
k_{FU}	Fog user master secret key
$k_{FS}^{(FU)}$	Secret key shared between FS and FU
k_s	Session key
$H(x)$	A hash invocation on input x
r_{FU}/r_{FS}	Random nonce picked by FU/FS
(pk_{RA}, sk_{RA})	Public/Private key pair of RA
(pk_{FS}, sk_{FS})	Public/Private key pair of FS
$E(k, x)$	Symmetric key encryption of x using key k
$D(k, x)$	Symmetric key decryption x using key k
$E_{pk}(x)$	public key encryption of x using key pk
$X \rightarrow Y$	X computes and sends to Y

strong symmetric encryption scheme (eg. AES), while H is a strong hash function (e.g. SHA-1,SHA-256, etc.). For simplicity we drop the subscript indexes since they are understood. The protocol consists of three phases: (i) Initialization phase, (ii) Registration phase and (iii) Authentication phase.

5.1 Initialization Phase

RA has her own public/private key pair (pk_{RA}, sk_{RA}), where pk_{RA} is known to all servers. Each server has his own public/private key pair (pk_{FS}, sk_{FS}) where RA stores pk_{FS} of each server FS. For each server, FS, in every Fog F, under the authority of RA, RA picks a unique identity ID_{FS} and sends it to FS signed with RA's signature key sk_{RA}. Notice that ID_{FS} is not secret.

5.2 Registration Phase

The registration phase is illustrated in Figure 3. A Fog user FU of identity ID_{FU} approaches the registration authority RA to register. The Fog network F has an identity ID_F and a set of Fog servers $\mathcal{FS}$ with each Fog server FS has an identity ID_{FS}. The registration is as follows:

- FU shows his identity ID_{FU} to the RA.
- RA picks a long-lived random master secret key (with long enough bit-length) k_{FU} for FU.
- FU stores $\langle ID_{FU}, k_{FU} \rangle$ on his smart device/card.
- For each $FS \in \mathcal{FS}$ in each F, RA computes the FS's secret key for FU as $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$ as shown in Figure 4 [2].
- For each server $FS \in \mathcal{FS}$, RA sends ID_{FU} and $k_{FS}^{(FU)}$ to FS encrypted under FS's public key, pk_{FS}. I.e. ID_{FU} and $E_{pk_{FS}}(k_{FS}^{(FU)})$. All signed with RA's signature key, sk_{RA} for authenticity.
- Finally, each server FS decrypts and stores the tuple $\langle ID_{FU}, k_{FS}^{(FU)} \rangle$ for each FU.

Remark (Joining of a new Fog server). We remark that, whenever a new Fog server (FS) joins a Fog, the RA runs the initialization phase for this server to setup a new identity ID_{FS} for this server and then computes the secret keys $k_{FS}^{(FU_j)} = H(ID_F, ID_{FS}, k_{FU_j})$ for all j of Fog users. We emphasize that this is done without the incorporation of the Fog users FU_j's who are already registered and without any increased overheads on the Fog users' side [3].

5.3 Authentication Phase

The authentication phase is illustrated in Figure 5. When a registered Fog user FU is in the location of the Fog F and needs to authenticate with a server FS, they proceed as follows (notice that, initially, FU does not know the identity ID_{FS} of any server. He just requests a Fog service) :

[2] Figure 4, shows why we named our scheme "Octopus": The master key k_{FU_j} represents the head of an octopus while the generated secret keys $k_{FS_i}^{(FU_j)}$ represent its arms. A user with the same head can later authenticate with any of the arms.

[3] It takes 0.006 ms for one hash invocation on Intel(R) Xeon(R) CPU E5520 @ 2.27G. Therefore, it takes less than a minute to generate FS-FU secret keys for 10 million Fog users.

FU	RA	FS
$\langle ID_{FU} \rangle$	$\langle ID_{FS}, pk_{FS}, sk_{RA} \rangle$	$\langle pk_{RA}, sk_{FS} \rangle$
$\xrightarrow{ID_{FU}}$		
	Picks k_{FU}, Stores $\langle ID_{FU}, k_{FU} \rangle$. $\xleftarrow{k_{FU}}$	
Stores $\langle ID_{FU}, k_{FU} \rangle$		
	Computes $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$, Computes $E_{pk_{FS}}(k_{FS}^{(FU)})$. $\xrightarrow{\langle ID_{FU}, E_{pk_{FS}}(k_{FS}^{(FU)}) \rangle}$ Signed with sk_{RA}	
		Verifies RA signature using pk_{RA}, Aborts if the verification fails, else, Decrypts using sk_{FS}, Stores $\langle ID_{FU}, k_{FS}^{(FU)} \rangle$ for this FU.

Figure 3: The registration phase of our scheme

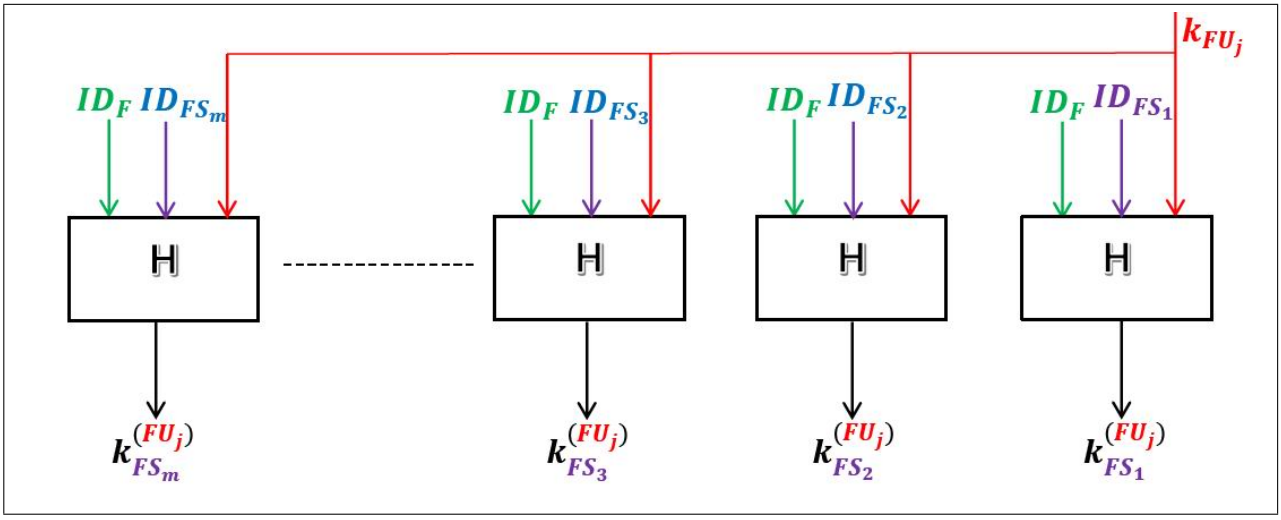


Figure 4: Generation of the FS-FU_j secret keys for user FU_j

$FU \rightarrow F$:

- Picks a random nonce r_{FU}.
- Broadcasts the tuple, $\langle HelloFog, ID_{FU}, r_{FU} \rangle$.

$FS \rightarrow FU$: An in-range server $FS \in \mathcal{FS}$:

- Checks that ID_{FU} is registered, else abort.
- Fetches $k_{FS}^{(FU)}$ for this ID_{FU}.
- Picks a random nonce r_{FS}.
- Prepares the encryption $E(k_{FS}^{(FU)}, (r_{FU}, r_{FS}))$, where $E(K, X)$ is a symmetric encryption of X using secret key K.
- Replies with the tuple, $\langle ID_{FS}, ID_F, ID_{FU}, E(k_{FS}^{(FU)}, (r_{FU}, r_{FS})) \rangle$.

$FU \rightarrow FS$:

- Using the received ID_{FS} and the stored k_{FU}, computes $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$.
- Decrypts and checks equality of r_{FU} with the received one. If the check fails then abort, otherwise,
- Picks a session key k_s, computes $E(k_{FS}^{(FU)}, (r_{FS}, k_s))$.
- Replies with the tuple, $\langle ID_{FS}, ID_F, ID_{FU}, E(k_{FS}^{(FU)}, (r_{FS}, k_s)) \rangle$.

FS:

- Using $k_{FS}^{(FU)}$, decrypts for (r_{FS}, k_s).

- Checks equality of r_{FS} with the received one, if the check fails then abort, otherwise accepts k_s as a session key.

6 Security Analysis

In this section we analyze the security of our scheme. First we show that the basic security requirements are satisfied, then we proceed to discuss the resistance of our scheme to common adversarial attacks. Finally we provide a formal security proof.

6.1 Basic Security Requirements

Mutual authentication. Mutual authentication between FU and FS is achieved, because both are able to deduce FU-FS secret key $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$, which is used to encrypt/decrypt for the session key k_s, $E(k_{FS}^{(FU)}, (r_{FS}, k_s))$ by FU and $D(k_{FS}^{(FU)}, (E(k_{FS}^{(FU)}, (r_{FS}, k_s)))$ by FS. The session key k_s will not be common to FU and FS unless the encryption and decryption are performed using the same secret key $k_{FS}^{(FU)}$. The Fog user FU generates the secret key $k_{FS}^{(FU)}$ locally, using his master secret key k_{FU} and the claimed server identity ID_{FS}. On the other hand, the RA has generated $k_{FS}^{(FU)}$ in the same way and delivered it secretly to the server. Hence, if a server identity ID_{FS} is claimed without knowing $k_{FS}^{(FU)}$, the server will not be authenticated by a legitimate user. On the other hand, a Fog user that does not hold the correct k_{FU} matching his identity ID_{FU} stored on the server, will not be authenticated by the server.

Protection of $k_{FS}^{(FU)}$. By inspecting our authentication protocol, the shared key $k_{FS}^{(FU)}$ is never used to encrypt a plaintext known to an eavesdropper, it is used to encrypt messages with a fresh random nonce r_{FS} as part of the plaintext. This random nonce is long, temporary, unknown to an eavesdropper and never placed on the channel in the clear.

Confidential communication session. The session key k_s is shared by both participants before performing their subsequent communication. The FU-FS secret key $k_{FS}^{(FU)}$ is known only to FU and FS, and is used to encrypt/decrypt for k_s. Therefore, the proposed scheme provides confidential communication.

Low computation and storage costs. There is no exponential computation or public key computation required on both sides during the authentication phase in the proposed scheme. Only a few hashes and symmetric encryptions/decryptions. Also, the scheme requires the user to store one master secret key k_{FU}, beside few short identities. Hence, the proposed scheme is efficient and easy to implement on smart cards. Therefore, the proposed scheme provides low computation complexity and storage complexity.

Simple key management. In the proposed scheme, the key management is very simple since, only the long-term secret key k_{FU} is maintained at the RA and on the FU's smart device. There is no PKI required at the FU. Furthermore, FS stores only one secret key for each registered user beside his short identity.

Session independence. The fresh session key k_s is not deduced from previous session keys, and there is no relationship among the session keys. Each session key is chosen as a fresh random string. Hence, a compromise of one session key does not affect other past/future sessions.

6.2 Adversarial Attacks

Fog server compromise (Rogue node). When an adversary corrupts/compromises a Fog server FS_i, then she knows all the FU-FS_i secret keys $\{k_{FS_i}^{(FU_1)}, ..., k_{FS_i}^{(FU_n)}\}$ of the users on this server. We emphasize the following:

- *This compromise does not threaten the security of the master secret key k_{FU} of any user FU given that the used hash function is a strong one-way function and the master secret key k_{FU} as an input to the hash function is long enough to withstand brute force given any compromised $k_{FS}^{(FU)}$.*

- *Compromising the server FS_i does not allow the adversary to deduce any other FU-FS secret keys on any other server on this Fog or any other server on any other Fog. This follows from the fact that, the FU-FS_i are generated independently by applying a one-way hash function on the master secret key k_{FU} of FU and the server's identity ID_{FS_i} since k_{FU} is not known to the adversary.*

The countermeasure for a server compromise is simple. Simply after the corrupted server is cleaned (rebooted, scanned, etc.) the registration authority RA chooses a new identity ID'_{FS_i} for this particular server, regenerates new set of FU-FS_i secret keys, $\{k'^{(FU_1)}_{FS_i}, ..., k'^{(FU_n)}_{FS_i}\}$ where $k'^{(FU_j)}_{FS_i} = H(ID_F, ID'_{FS_i}, k_{FU_j})$ and sends them to FS_i as in the registration phase. The Fog users are informed with the rogue identity ID_{FS_i} in public. *We emphasize that FU master secret key k_{FU} is safe and that FU is not required to incorporate in any new registration processes.*

Secret key guessing attacks. The only secret on the user's side is the user's master key k_{FU}. The key is a strong secret key with long enough bits (to protect against brute force attacks in case a server is compromised) and protected in a tamper-resistant mechanism, such as a smart card. There is no efficient way to obtain it, but brute-force guessing. Therefore,

FU $\langle ID_{FU}, k_{FU}\rangle$		FS $\langle ID_{FU}, ID_{FS}, ID_F, k_{FS}^{(FU)}\rangle$
Picks r_{FU}		
	$\xrightarrow{\langle HelloFog, ID_{FU}, r_{FU}\rangle}$	
		Checks that ID_{FU} exists Fetches $k_{FS}^{(FU)}$ Picks r_{FS} Computes the encryption, $E(k_{FS}^{(FU)}, (r_{FU}, r_{FS}))$
	$\xleftarrow{\langle ID_{FS}, ID_F, ID_{FU}, E(k_{FS}^{(FU)}, (r_{FU}, r_{FS}))\rangle}$	
Computes, $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$ Decrypts & checks validity of r_{FU} Aborts if the check fails, otherwise, Picks k_s Computes $E(k_{FS}^{(FU)}, (r_{FS}, k_s))$		
	$\xrightarrow{\langle ID_{FS}, ID_F, ID_{FU}, E(k_{FS}^{(FU)}, (r_{FS}, k_s))\rangle}$	
		Decrypts for r_{FS}, k_s Checks validity of r_{FS} Aborts if the check fails, else, Accepts k_s as a session key

Figure 5: Edge-Fog mutual authentication phase

the proposed scheme is secure against secret key guessing attacks.

Replay/impersonation attacks. Consider the case where an adversary records all data transfered between FU and FS, during the authentication phase and the whole session. Now an adversary may try to replay any message at any round, wishing to succeed in the replay and impersonate either FU or FS. Of course trying to replay a data session, encrypted under an old session key k_s will not succeed, unless the adversary succeeds in the authentication phase. Now, lets see what happens if the adversary replays each round in the authentication phase. Assuming that the adversary is a Fog user FU', that replays $\langle HelloFog, ID_F, ID_{FU}, r_{FU}\rangle$ in Figure 5. The server FS replies with the tuple, $\langle ID_{FS}, ID_F, ID_{FU}, E(k_{FS}^{(FU)}, (r_{FU}, r_{FS}))\rangle$ challenging FU' with the random nonce r_{FS}. FU' does not know $k_{FS}^{(FU)}$ and hence, he performs the encryption using some random k^* as $E(k^*, R)$ for some R. Now, the server FS decrypts using the correct $k_{FS}^{(FU)} \neq k^*$, resulting in some $r_{FS}^* \neq r_{FS}$ and hence, the r_{FS} produced by FS equals the received one only with negligible probability. Hence, FU' will not succeed in the third round. In the same way, replaying the second round by an adversary FS', impersonating FS, will not succeed due to the random fresh challenge r_{FU}. Finally, trying to impersonate FU and replay a previously recorded third round, will not succeed since the included random nonce does not match that of round two by FS.

Man-in-the-middle attack. Consider an adversary that puts herself as an intermediate node between FU and FS. This adversary does not know k_{FU}. Now, this adversary tries to masquerade each party to the other. Since, this adversary does not know k_{FU}, she cannot deduce the secret key $k_{FS}^{(FU)}$, generated locally by FU and stored in FS. The adversary cannot generate any correct encryptions of the random nonce r_{FS} or r_{FU}. Hence, the man-in-the-middle attack fails.

Fog user compromise. If FU device is compromised, then the master secret key k_{FU} falls in the hands of the adversary. *This compromise does not affect any other Fog user.* However, for this compromised user FU, he must report to the RA for revoking his compromised master key and register for a new one.

6.3 Formal Security Proof

In this subsection, we give a formal proof of the security of our scheme.

Theorem 1. *Assuming E and H used in our authentication scheme are secure pseudo-random function families, then our scheme based on E and H is a secure mutual entity authentication and key exchange protocol.*

Proof. Assuming H is a strong hash function and that k_{FU} is a long enough secret key, it is infeasible for an attacker knowing any $k_{FS}^{(FU)} = H(ID_F, ID_{FS}, k_{FU})$ to reach k_{FU}. Now we continue to prove our theorem using BAN logic [6] given that both FU and FS believe in $k_{FS}^{(FU)}$ as follows:

Idealization. By removing all plaintext messages, the idealized messages between FU and FS in our scheme are as follows:

- M1: $FU \rightarrow FS$: $-$
- M2: $FS \rightarrow FU$: $\{(r_{FU}, r_{FS})\}_{k_{FS}^{(FU)}}$
- M3: $FU \rightarrow FS$: $\{r_{FS}, FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS)\}_{k_{FS}^{(FU)}}$

Assumptions. The assumptions of the protocol are as follows:

- A1: $FU \mid\equiv \#(r_{FU})$
- A2: $FS \mid\equiv \#(r_{FS})$
- A3: $FU \mid\equiv (FU \overset{k_{FS}^{(FU)}}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$
- A4: $FS \mid\equiv FS \overset{k_{FS}^{(FU)}}{\longleftrightarrow} FU$
- A5: $FS \mid\equiv (FU \Rightarrow FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$
- A6: $FU \mid\equiv (FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$

Main goals.

- G1: $FS \mid\equiv (FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$
- G2: $FS \mid\equiv FU \mid\equiv (FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$.
- G3: $FU \mid\equiv FS \mid\equiv r_{FU}$
- G4: $FS \mid\equiv FU \mid\equiv r_{FS}$

Analysis.

From assumptions A1 and A3 and message M2,

$$\frac{FU \mid\equiv \#(r_{FU})}{FU \mid\equiv \#(r_{FU}, r_{FS})} \text{ (Freshness rule)}$$

$$\frac{FU \mid\equiv FU \overset{k_{FS}^{(FU)}}{\longleftrightarrow} FS, FU \lhd \{(r_{FU}, r_{FS})\}_{k_{FS}^{(FU)}}}{FU \mid\equiv FS \mid\sim (r_{FU}, r_{FS})}$$

(Message meaning rule)

$$\frac{FU \mid\equiv \#(r_{FU}, r_{FS}), FU \mid\equiv FS \mid\sim (r_{FU}, r_{FS})}{FU \mid\equiv FS \mid\equiv (r_{FU}, r_{FS})}$$

(Nonce verification rule)

$$\frac{FU \mid\equiv FS \mid\equiv (r_{FU}, r_{FS})}{FU \mid\equiv FS \mid\equiv r_{FU}} \text{ (Belief rule)}$$

This satisfies goal G3.

Let $X = (FU \overset{k_s}{\longleftrightarrow} FS, \#(FU \overset{k_s}{\longleftrightarrow} FS))$. From assumptions A2 and A4 and message M3, we have,

$$\frac{FS \mid\equiv \#(r_{FS})}{FS \mid\equiv \#(r_{FS}, X)} \text{ (Freshness rule)}$$

$$\frac{FS \mid\equiv FU \overset{k_{FS}^{(FU)}}{\longleftrightarrow} FS, FS \lhd \{(r_{FS}, X)\}_{k_{FS}^{(FU)}}}{FS \mid\equiv FU \mid\sim (r_{FS}, X)}$$

(Message meaning rule)

$$\frac{FS \mid\equiv \#(r_{FS}, X), FS \mid\equiv FU \mid\sim (r_{FS}, X)}{FS \mid\equiv FU \mid\equiv (r_{FS}, X)}$$

(Nonce verification rule)

$$\frac{FS \mid\equiv FU \mid\equiv (r_{FS}, X)}{FS \mid\equiv FU \mid\equiv X} \text{ (Belief rule)}$$

Thus, goal G2 is reached.

$$\frac{FS \mid\equiv FU \mid\equiv (r_{FS}, X)}{FS \mid\equiv FU \mid\equiv r_{FS}} \text{ (Belief rule)}$$

This satisfies goal G4.

From assumption A5 we have,

$$\frac{FS \mid\equiv FU \Rightarrow X, FS \mid\equiv FU \mid\equiv X}{FS \mid\equiv X} \text{ (jursdiction rule)}$$

Thus, goal G1 is reached and so the proof of the theorem. □

7 Complexity Evaluation

Our protocol uses only two simple cryptographic primitives; several invocations of a strong hash function H and symmetric encryption/decryption E/D (e.g. AES), making the protocol very efficient for smart card implementation. There are many hash functions out there for cryptographic applications such as SHA-1, SHA-2, SHA-224, SHA-256, etc. [27]. For the concrete evaluation of the complexity of our protocol, we assume SHA-1 as the hash function in place. SHA-1 takes an input as an arbitrary length message partitioned in blocks of 512 bits where the last block is padded with zeros to complete the block size. Each 512-bit block produces a SHA-1 output of 160 bits where these 160 bits are re-invoked as input with the next 512-bit message block. The final output of SHA-1 is 160 bits as the hash of the arbitrary length message [27].

We assume the identity ID_{FU} is of size 3 bytes (enough for a huge population, however the size maybe a little longer since the user's identity contains printable characters). The identities ID_{FS} and ID_F are assumed one byte each. These choices may differ according to the population.

We also assume that k_{FU} is of 160 bits just for the purpose of evaluation. The length of k_{FU} may be chosen freely by RA, since it is not incorporated in any symmetric key encryptions. k_{FU} is used only as an input to the hash function to generate the keys $k_{FS}^{(FU)}$, therefore, its effective bit-length is exactly its actual bit-length chosen freely be RA.

7.1 Storage Requirements

The storage requirements of our scheme are given in Table 2 and detailed next.

Fog user FU**.** The Fog user FU is required to store the tuple $\langle ID_{FU}, k_{FU} \rangle$ which is a random long enough string as his master secret key k_{FU} in addition to

Table 2: Storage and computations requirements of our scheme

	Storage	Computations		
		Initialization phase	Registration phase	Authentication phase
FU	One short ID. One secret key.	–	–	One hash invocation. One symmetric encryption. One symmetric decryption.
FS	Two public keys. One private key. One secret key/FU. One short ID/FU. One short ID_F. One short ID_{FS}.	One signature verification.	One signature verification. One private key decryption.	One symmetric encryption. One symmetric decryption.
RA	One public key. One private key. One public key/FS. One short ID/FS. One short ID/F. One secret key/FU. One short ID/FU.	One signature generation. One hash invocation/FS.	One private key encryption. One hash invocation. One signature generation.	–

a one short string as his identity ID_{FU}.

Fog server FS**.** FS is required to store a short string as his identity ID_{FS} and the tuple $\langle ID_{FU}, k_{FS}^{(FU)} \rangle$ for each FU which consists of a short string as ID_{FU} and the FS-FU secret key $k_{FS}^{(FU)}$. This is in addition to RA's public verification key pk_{RA} and his own public/private key pair (pk_{FS}, sk_{FS}) of the public key cryptosystem in place.

Registration authority RA**.** RA stores the master secret keys of all registered Fog users in addition to the public keys of all Fog servers and her own public/private key pair (pk_{RA}, sk_{RA}) of the public key cryptosystem in place. These are in addition to the users and servers short identities.

7.2 Computation Complexity

The computation complexity of our scheme is given in Table 2.

Fog user FU**.** By inspecting our protocol, FU does not perform any computations in the registration phase, he just receives k_{FU}. In the authentication phase, FU performs only one hash invocation, one symmetric encryption and one symmetric decryption.

Fog server FS**.** In the registration phase, FS performs one signature verification for his identity ID_{FS} and one signature verification for each registered user. In the authentication phase, performs one symmetric encryption and one symmetric decryption.

Registration authority RA**.** In the Initialization phase, performs one hash invocation for the generation of ID_{FS} and one digital signature on ID_{FS} for each FS. In the registration phase, for each registered user FU and each Fog server FS, performs one hash invocation, one digital signature and one public key encryption.

7.3 Computation Time

A simulation hardware environment is setup to measure the computation time of the cryptographic primitives. The simulation environment is a 32-bit Cortex-M3 microcontroller with 72 MHz ARM MCU and 512 KB memory [21]. A secret key encryption of an AES-128 block cipher takes 0.919 ms, while the decryption takes 1.074 ms. A one invocation of hash function SHA-1 takes 0.06 ms .

It takes FU about 0.06 ms (one SHA-1 invocations) plus 0.919 ms (one AES encryption) plus 1.074 ms (One AES decryption) totaling 2.053 ms on FU side.

On the FS side, it takes about 0.919 ms (One AES encryption) plus 1.074 ms (One AES decryption) totaling 1.993 ms. Computation time is summarized in Table 3.

7.4 Energy Consumption

In this part, the energy consumption consumed by cryptographic operations is used to evaluate the schemes. This time, we use a low-processor and 64 MB memory running Windows Mobile 5.0 for pocket pc. According to PXA270, the typical power consumption of PXA270 in active is 570 mW [4]. Therefore, using the computation time in the previous subsection, we can calculate the corresponding energy consumption. For example, if it takes 0.919 ms to complete a AES-128, the energy consumption is approximately $0.919*(570/1000) = 0.523$ mJ. So the energy consumed by FU is 1.17 mJ while the energy

[4] http://pdf.dzsc.com/CXX/NHPXA270Cxxx.pdf

consumed by FS is 1.14 mJ. The energy consumed by our scheme is summarized in Table 3.

7.5 Comparison with Closely Related Work

Although, up to the time this paper was written, and up to our knowledge, there is no contribution that directly targets the mutual authentication in the Edge-Fog-Cloud architecture, we discuss other close contributions targeting wireless sensor networks. The roaming protocols of [9, 29] used the identity-based cryptography and group signature to realize the local authentication of the roaming protocol. The communication times of the mobile node in their protocols do not contain the transmission of the authentication materials. The communication times of [10] are equal or greater than by four times, because of the re-authentication process after every moving. The protocol stores all the authentication materials into the neighboring nodes through broadcast, and the broadcast communication computes at least once communication.

These protocols and the recent in [40] employ public-key cryptosystems and bilinear pairings as essential requirements which dramatically increase the computations complexities specially for smart cards. Our protocol does not require the engagement of public key cryptosystems.

Table 3: Computation time and energy consumption

	Computation time		**Energy consumed**
	Round 1	**Round 2**	
FU	–	2.053 ms	1.17 mJ
FS	0.919 ms	1.074 ms	1.14 mJ

8 Discussions

Some applications, such as vehicle-to-vehicle communications in VANETS [32, 30, 31], requires that the Fog users (vehicles) interact with each other within a certain Fog. Given that, each Fog user FU_j shares a secret key with a Fog server FS as $k_{FS}^{(FU_j)}$, there are many protocols that allow this server to establish a common session key for these users allowing them to communicate in a private way. For example, one may consider the Wide-Mouth-Frog protocol [6]. Another protocol is the Needham-Schroeder Symmetric Key Protocol based on a symmetric encryption algorithm, which forms the basis for the Kerberos protocol [25]. Many other server-based key distribution exist [5].

Anonymity is one of the important services that must be available to users in the digital world as long as they behave honestly. Users' communication must be kept authenticated and anonymous unless malicious behaviors are detected. In this case the accused user's clear identity must be traced and revealed by the system to solve accusations [14, 13, 15, 12]. In the Edge-Fog-Cloud model, Edge users have the right to keep their identities anonymous as long as they are honest, while on the other hand, the CSP has the right to be able to trace any user to his clear identity once he/she misbehaves. So, it would be a nice open problem to find a way to add this service to our scheme, or to devise a new authentication scheme that provides this service to the Edge users.

An important requirement of anonymity by many applications is unlinkability of virtual identities, i.e. an adversary $\mathcal{A}$ must not be able to link activities (e.g. transactions) to the same person/entity although his clear identity is blinded from $\mathcal{A}$. There exist schemes based on what is known as "temporary identities" (e.g. [26, 42]). In such schemes, the user shares his identity with the server and this identity is updated to a new fresh string after each session in a way unpredictable to the attacker. Such schemes are computationally efficient and secure. However, the problem with such schemes is that, both the user and the server must be in synchronism with the current temporary identity. At a certain round of the protocol, the adversary may disrupt the communication (e.g. through jamming) resulting in a loss of synchronism between the user and the server. The consequences of such attack is the DoS of the current session and all future sessions. These schemes may be suitable for small area networks where it is easy to reset and reinitialize the system when such attack is detected. However, for large scale networks such as the Edge-Fog-Cloud architecture, such schemes are impractical.

9 Conclusions

Services of Fog computing are offered to massive-scale end users where it is hard to realize PKI on such a large scale at the Edge of the network. We proposed a secure and efficient scheme to allow any Fog user to mutually authenticate with any Fog server in any Fog under the authority of a Cloud service provider. Our Scheme does not require a Fog user to be incorporated in any PKI. The Fog user is required to store one master secret key in the registration phase only once. Using this master key the Fog user is able to mutually authenticate with any Fog server managed by the Cloud service provider. On the other hand, Our scheme provides a simple countermeasures if one or more Fog servers are compromised and fully corrupted by an adversary. Even if all the Fog servers are corrupted, the master secret key of the user with long enough bit-length remains secure against brute force and hence, the Fog user does not need to be incorporated in any re-initialization or re-registration of a new master key. Also, the Fog user is able to mutually authenticate with any Fog server that joins a Fog after Fog user registration without the need for the user to re-register and without any extra overheads on the user's side.

Our Scheme is computationally efficient, even in the existence of huge population. It requires the Fog users and the Fog servers to perform very few hash invocations

and symmetric key encryptions/decryptions and hence, it is suitable for implementation on smart cards and devices with limited resources.

References

[1] M. Armbrust, A. Fox, R. Griffith, et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50–58, Apr. 2010.

[2] D. Balfanz, D. K Smetters, P. Stewart and H. C. Wong, "Talking to strangers: Authentication in adhoc wireless networks," in *Symposium on Network and Distributed Systems Security*, 2002.

[3] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the ACM Workshop on Mobile Cloud Computing*, pp. 13–16, New York, USA, 2012.

[4] S. Bouzefrane, B. Mostefa, F. Amira, F. Houacine and H. Cagnon, "Cloudlets authentication in NFC-based mobile computing," in *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 267–272, 2014.

[5] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer Science & Business Media, 2013.

[6] 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.

[7] E. Damiani, D. Capitani D. Vimercati, S. Paraboschi, P. Samarati and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 207–216, 2002.

[8] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, 2011.

[9] D. He, J. Bu, S. Chan, C. Chen and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Transactions on Wireless Communications*, vol. 10, no. 2, pp. 431–436, 2011.

[10] D. He, C. Chen, S. Chan and J. Bu, "Strong roaming authentication technique for wireless and mobile networks," *International Journal of Communication Systems*, vol. 26, no. 8, pp. 1028–1037, 2013.

[11] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung and J. H. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.

[12] M. H. Ibrahim, "AATCT: Anonymously authenticated transmission on the cloud with traceability," *International Journal of Advanced Computer Science & Applications*, vol. 6, no. 9, pp. 251–259, 2015

[13] M. H. Ibrahim, "Resisting traitors in linkable democratic group signatures," *International Journal of Network Security*, vol. 9, no. 1, pp. 51–60, 2009.

[14] M. H. Ibrahim, "Noninteractive, anonymously authenticated, and traceable message transmission for VANETs," *International Journal of Vehicular Technology*, 2009.

[15] M. H. Ibrahim, "Secure anonymously authenticated and traceable enterprise DRM system," *International Journal of Computer Applications*, vol. 126, no. 3, September 2015.

[16] A. Jøsang, R. Ismail and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[17] M. Kumar, "An enhanced remote user authentication scheme with smart card," *International Journal of Network Security*, vol. 10, no. 3, pp. 175–184, 2010.

[18] M. Kumar, "A new secure remote user authentication scheme with smart cards," *International Journal of Network Security*, vol. 11, no. 2, pp. 88–93, 2010.

[19] C. C. Lee, C. H. Liu and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[20] M. Li, W. Lou and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 51–58, 2010.

[21] J. Liu, Q. Li, R. Yan and R. Sun, "Efficient authenticated key exchange protocols for wireless body area networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1–11, 2015.

[22] R. Lu, Z. Cao, Z. Chai and X. Liang, "A simple user authentication scheme for grid computing.," *International Journal of Network Security*, vol. 7, no. 2, pp. 202–206, 2008.

[23] R. Lu, X. Li, X. Liang, X. S. Shen and X. Lin, "Grs: The green, reliability, and security of emerging machine to machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 28–35, 2011.

[24] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.

[25] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[26] C. Noureddine, F. Cherif, P. L. Cayrel and B. Mohamed, "Improved rfid authentication protocol based on randomized mceliece cryptosystem," *International Journal of Network Security*, vol. 17, no. 4, pp. 413–422, 2015.

[27] National Institute of Standards and Technology (NIST), "Fips 180-4, secure hash standard," *Federal Information Processing Standards Publication.*

[28] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[29] Y. Qiu, J. Zhou, J. Baek and J. Lopez, "Authentication and key establishment in dynamic wireless sensor networks," *Sensors*, vol. 10, no. 4, pp. 3718–3731, 2010.

[30] F. M. Salem, M. H. Ibrahim and I. I. Ibrahim, "Non-interactive authentication scheme providing privacy among drivers in vehicle-to-vehicle networks," in *IEEE Sixth International Conference on Networking and Services*, pp. 156–161, 2010.

[31] F. M. Salem, M. H. Ibrahim and I. I. Ibrahim, "Non-interactive secure and privacy preserving protocol for inter-vehicle communication networks," in *IEEE Seventh International Conference on Information Technology: New Generations*, pp. 108–113, 2010.

[32] F. M. Salem, M. H. Ibrahim and I. I. Ibrahim, "Efficient noninteractive secure protocol enforcing privacy in vehicle-to-roadside communication networks," *International Journal of Vehicular Technology*, vol. 2012, 2012.

[33] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *IEEE Federated Conference on Computer Science and Information Systems*, pp. 1–8, 2014.

[34] J. L. Tsai, "Efficient nonce-based authentication scheme for session initiation protocol," *International Journal of Network Security*, vol. 9, no. 1, pp. 12–16, 2009.

[35] J. Valenzuela, J. Wang and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1052–1062, 2013.

[36] Z. Wan, K. Ren and B. Preneel, "A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks," in *Proceedings of the First ACM conference on Wireless Network Security*, pp. 62–67, 2008.

[37] J. Wang, Y. Yanshuo and K. Zhou, "A regular expression matching approach to distributed wireless network security system," *International Journal of Network Security*, vol. 16, no. 5, pp. 382–388, 2014.

[38] W. Wang and Z. Lu, "Survey cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, pp. 1344–1371, April 2013.

[39] C. Wei, Z. M. Fadlullah, N. Kato and I. Stojmenovic, "On optimally reducing power loss in microgrids with power storage devices," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 7, pp. 1361–1370, 2014.

[40] Q. Q. Xie, S. R. Jiang, L. M. Wang and C. C. Chang, "Composable secure roaming authentication protocol for cloud-assisted body sensor networks," *International Journal of Network Security*, vol. 18, no. 5, pp. 816–831, 2016.

[41] G. Yang, Q. Huang, D. S. Wong and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 168–174, 2010.

[42] E. J. Yoon, K. Y. Yoo, J. W. Hong, S. Y. Yoon, D. I. Park and M. J. Choi, "An efficient and secure anonymous authentication scheme for mobile satellite communication systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–10, 2011.

Maged Hamada Ibrahim. B.Sc. in Communications and Computers Engineering from Helwan University with Distinction and Honor's Degree in 1995. He also obtained his M.Sc. from the same University in 2001. Then his Ph.D. from Helwan University in 2005. He is now an Associate Professor at Helwan University. He is joining several network security projects in Egypt. His main interest is engineering cryptography and communications security. More specifically, working on the design of efficient and secure cryptographic algorithms and protocols, in particular, secure distributed multiparty computations, public key infrastructures, digital signatures, digital rights management protocols and non-cryptographic solutions to telecommunication security problems. Other things that interest him are number theory and the inspection of mathematics for designing secure and efficient cryptographic schemes.

Anti-fake Digital Watermarking Algorithm Based on QR Codes and DWT

Jiaohua Qin, Ruxin Sun, Xuyu Xiang, Hao Li and Huajun Huang
(Corresponding author: Jiaohua Qin)
College of Computer and Information Engineering, Central South University of Forestry and Technology
Changsha 410004, China
(Email: qinjiaohua@163.com)
(Received Sep. 16, 2015; revised and accepted Nov. 15 & Nov. 30, 2015)

Abstract

This article proposes an anti-fake QR codes watermarking algorithm based on the DWT and SVD, aiming at the security problem of QR code in the actual application. Firstly, this paper analyses the advantage of QR code as well as its problems. Secondly, the principle of the chaos encryption and discrete wavelet transform are introduced in detail. Then, we design a new watermarking barcode by the combination of chaos encryption and singular value decomposition and discrete wavelet transform. Experimental results show that the proposed method is significantly superior to the prior arts on the anti-fake performance and watermarking quality.

Keywords: Chaotic encryption, discrete wavelet transform, QR code, watermark barcode

1 Introduction

In the research of 2D barcode, QR code with the advantages of strong error correcting ability, large capacity, being identified easily, becomes a more outstanding member in the barcode family, and it is also widely used. Except the characteristics that other 2D barcodes have, QR code has the advantages of high reliability, Chinese characters and image information representation ability, confidentiality and security, high reading speed, big data density, small occupied space and full reader. Therefore, QR codes are widely concerned interiorly, becoming a hot research and application of two dimensional barcode [2]. However, just like other barcode, the opening coding mode makes it perform badly in privacy as the lack of security methods in the strict sense.

According to "The global mobile phone security report in the first quarter of 2014", a total of 41199 models of the mobile malware were killed with a year-on-year growth of 63.9% [12]. The two-dimensional code technology has become a new channel of mobile phone viruses and phishing web site communication.In order to get rid of the hidden trouble of safety and protect the security of the information about the users, the two-dimensional code technology and information security technology must be combined to research and develop a safe and reliable two-dimensional code.

Spatial domain watermarking is an edge pixel expansion or reduction of the depth graphics module of the two-dimensional code. The use of two-dimensional code recognition algorithm for depth graphics module allows a certain error, and there exists many reservations module and no coding modules [13]. Coding these modules will not affect the correct recognition of two dimensional code, and can get the realization of embedded secret information. He et al. [5] proposed QR code digital watermarking method based on the least significant bit and its improved algorithm. The scheme embedded watermark into the least significant bit of QR code, its improved algorithm was for gray image. Because the QR code itself was binary image, the robustness of watermark bar code was poor as the embedding of the LSB, so it is difficult to extract the watermark when suffering attack. Zhu et al. [16] proposes the matrix coding based on the LSB algorithm, which can reduce the bits of the least significant bit that need to be modified. Since the scheme is based on the LSB algorithm of two-dimensional codes, so the robustness of the algorithm performs poor. Xie et al. [14] used chaotic mapping to control the position of watermark embedding QR code, which adapt strategy of the chaotic key adaptive adjustment, improved the capacity and robustness of watermark embedding. Since the scheme uses a key adaptive strategy, the watermark embedding process is not stable, needs repeated embedding and verification, and when the watermark information gets too large, chaotic key adjustment times and the algorithm consuming time will increase.

With the spreading spectrum and mapping for hidden information, Chao et al. [1] hide information by the strip and the space of the fine-tuning code according to the structural characteristics of two-dimensional barcode. G. Prabakaran et al. [9] extracted I component of the

video and did SVD decomposition by using singular value decomposition and discrete wavelet transform technology, then inserted the logo into a diagonal matrix of SVD decomposition, at last the video watermark logo can be obtained after reverse changes. Liu et al. [6] did DCT block transformation to vector images by using the discrete cosine transform and singular value decomposition of matrix, then did SVD decomposition on the coefficient matrix obtained by transformation, then did watermark embedding in the diagonal matrix. The algorithm has good invisibility and robustness, but it is complex and difficult to implement.

In this paper, we proposes an anti-fake QR codes watermarking algorithm based on the DWT and SVD. In Section 2, we introduce the detail algorithm of watermark barcode based on QR and DWT, and the experimental results and analysis is shown in Section 3. The conclusion is given is Section 4.

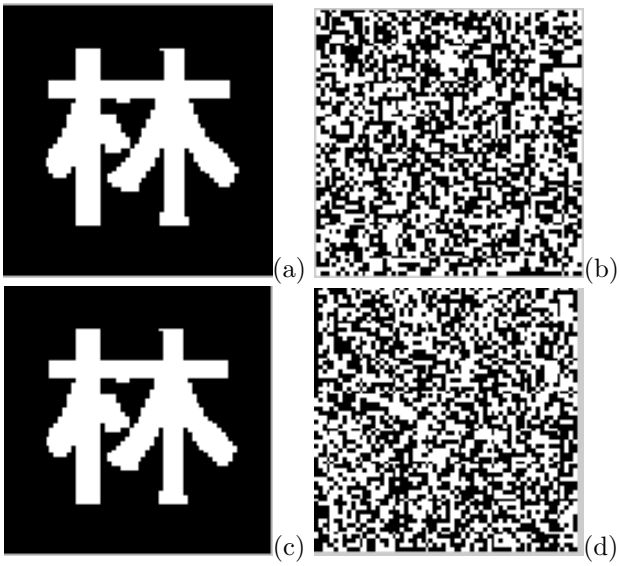


Figure 1: The watermarked image after using chaos encryption and decryption: (a) Original image, (b) The chaotic encrypted image, (c) Decryption image, (d) The error decrypting image

2 Watermark Barcode Based on QR and DWT

This paper took the QR code as the carrier, the binary image as the watermark information. Firstly, we encrypted watermark information by chaotic encryption. Secondly, we did 3 layers of the discrete wavelet transform to the QR code, then we went on with the singular value decomposition on diagonal components of high frequency after the wavelet transform. The final, we embedded chaotic encryption watermark information into the obtained diagonal matrix.

2.1 Chaotic Encryption

Chaos is a kind of complex dynamical behavior with special properties. It has the characteristics of extreme sensitivity to initial conditions and system parameters, movement track irregularity, intrinsic randomness, boundness, ergodicity. Therefore we can construct the encryption system by these characteristics [15].

Encryption system is very sensitive to initial value and parameters, it can provide a set of keys, and fully meet the demand of chaotic system tested by the cryptographic binary sequence. The uniform distribution of 0 and 1 satisfied the random numbers requirements, can be regarded as a random sequence. Stream cipher includes the chaos encryption and it is ineffective for block cipher attack method. Due to the unidirectional and the iterative of chaotic signal processing, the operated key stream is almost impossible to infer for the chosen plain text and cipher text attack method.

This paper uses chaos mapping to generate a chaotic sequence and transforms it into the dual-value matrix with the same size of watermarking, and do XOR operation on watermark to get the watermark encryption. In order to enhance the security of the watermark and the robustness, we can also carry out the scrambling operation on watermark. The encryption algorithm is as follows:

Step 1. Input the encrypted initial value.

Step 2. Generate a chaotic sequence with the same size of watermark by chaotic mapping formula. The adapted chaotic equation is as follows:

$$\begin{aligned} L(1) &= key, key \in (0,1) \\ L(i) &= 1 - 2 \times L(i-1) \times L(i-1), \\ i &= 2, 3 \cdots m \times n \end{aligned}$$

Where key is the initial key, $m * n$ is the size of the watermark image.

Step 3. Converted the generated sequence into 0, 1 sequences.

$$\begin{cases} L'(i)=1, L(i) \geq 0 \\ L'(i)=0, L(i) < 0 \end{cases}$$

Step 4. Do XOR operation on the generated 0, 1 sequence and the watermark.

Decryption is the reverse process of encryption. Decryption process needs to know the secret key, chaotic equation and the encrypted watermark image. The chaotic sequence is obtained through the chaotic formula and the secret key. Convert the chaos sequence into 0, 1 sequence and XOR with the encrypted watermark image, then image can be decrypted. If we use an error secret key, we will not get the decrypted watermarking image. The image chaotic encryption and decryption are shown in Figure 1.

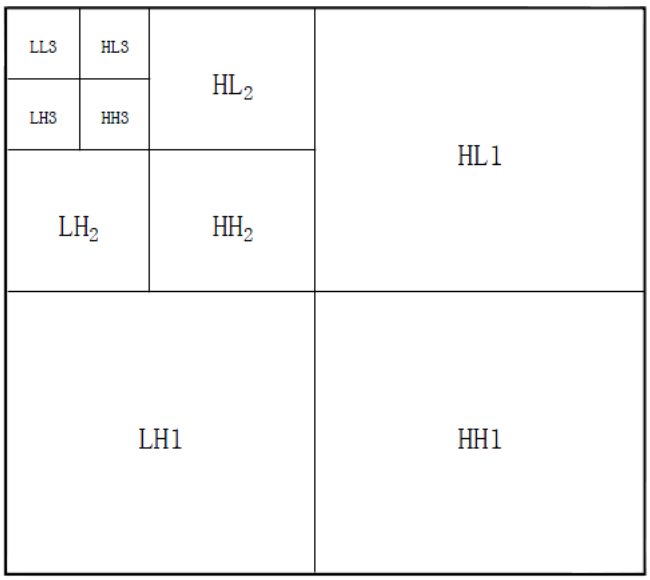


Figure 2: Decomposition diagram based on DWT

2.2 The Discrete Wavelet Transformation

The discrete wavelet transformation (DWT) means the discretization of the expansion factor a in the discretization continuous wavelet function and the translation factor b [10]:

$$a = a_0^m (a_0 > 1), b = nb_0 a_0^m (b_0 \in R, (m,n) \in Z^2) \quad (1)$$

Then

$$\varphi_(m,n)(t) = a_0^{m/2} \varphi(a_0^m t - nb_0)$$

In general, $a_0 = 2, b_0 = 1$:

$$\varphi_(m,n)(t) = 2^{m/2} \varphi(a^m t - nb)$$

For the discrete wavelet transform for arbitrary function $\varphi(t) \in L^2(R)$:

$$W_f(m,n) =< f, \varphi_{m,n} >= \int_{-\infty}^{+\infty} f(t) \times \overline{\varphi_{m,n}(k)}$$

If $f(t)$ is discrete, recorded as $f(k)$, then:

$$w_f(m,n) = \sum_k f(k) \times \overline{\varphi_{m,n}(k)}$$

From the wavelet multi-resolution and decomposition of the image signal characteristics, the principle of wavelet transform is in accordance with the octave to separate the signal spectrum, and the obtaining final signal is a low frequency sub-band in these octave band and several high frequency sub-band data [8].

Figure 2 is the multi resolution wavelet decomposition, and it is the decomposition figure after 3 times discrete wavelet transform. After the 3 times discrete wavelet transform, the LL_3 band is the low frequency sub-band. HH_K, LH_k, HL_k $(k = 1, 2, 3)$ and several other bands is the high frequency sub-band. The HL_k band is a sub-band obtained by going through a low-pass filtering firstly in the row direction on the upper level low frequency sub-band, and then a high-pass filteringin the column direction. So, the HL_k band mainly contains information of details of the signal in the horizontal direction on the vertical direction, HL_k is the horizontal detail sub-band. In the third layer wavelet decomposition, the low frequency sub-band LL_3 contains the lowest resolution information of the original images. HL_3, LH_3, HH_3 are the fine information data of LL_3. Because of the characteristics of multi-resolution decomposition of wavelet transform, wavelet analysis of the image has a very good directional selectivity, and can combine with the human visual system very well.

2.3 Watermarking Algorithm Based on DWT and QR Code

In this paper, watermark embedding algorithm is based on DWT and singular value decomposition. The watermark information after processing will be embedded into the three layer wavelet transformed the diagonal components, which reduces the influence from the image watermarking to the QR code.

2.3.1 The Watermark Embedding

This paper selects the QR codes as the carrier image of watermark embedding. The watermark information is a binary image. QR codes are generated by software. The watermark embedding process is shown in Figure 3:

The specific embedding procedure is as follows:

Step 1. Generate the QR code image to do discrete wavelet transform. Do three level discrete wavelet transforms to the QR code image. Get the parameters of LL_3, LH_3, HL_3, HH_3.

Step 2. Singular value decomposition to high frequency diagonal coefficient HH_3. On the HH_3 singular value decomposition, we can get the transformation matrix U, V and diagonal matrix D. D will be regard as the embedding position of the watermark.

Step 3. Chaotic encryption of watermark image W. By the singular value decomposition of chaotic encrypted watermark image, we will get the maximum singular values of watermark image, used to determine the embedding factor.

Step 4. The watermark embedding. The watermark should be embedded into the high frequency diagonal coefficient HH_3 of the QR code according to the embedding factor.

$$HH_3 w(i,j) = HH_3(i,j) + \alpha \times$$
$$HW(mod(i-1, wm) + 1, mod(j-1, wn) + 1)$$

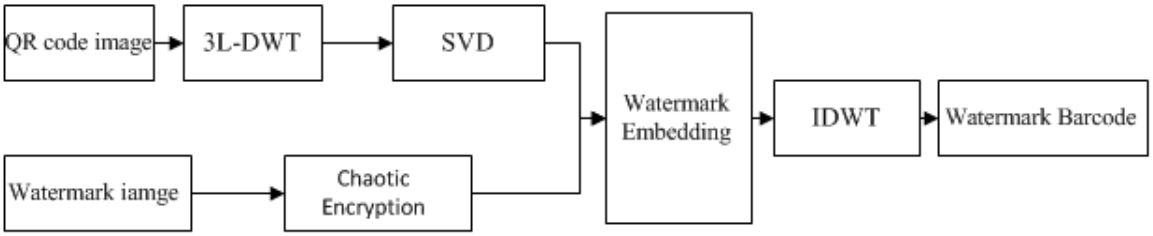


Figure 3: The flow-process diagram of watermark

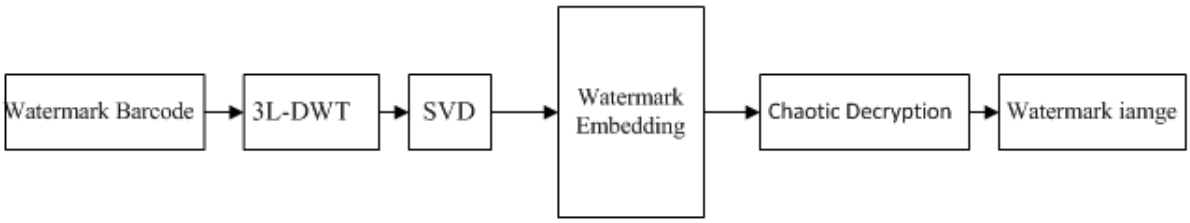


Figure 4: The watermark embedding process diagram

Among them, α is the embedding factor. We value it 0.01 and wn is the size of watermark image.

Step 5. SVD transformation on the obtained matrix. By the SVD transformation to Watermark embedded diagonal matrix D, we will obtain matrix U_1, V_1 and the diagonal matrix D_1. Do inverse SVD transformation on the first SVD transformed matrix U, V and the second SVD transformed diagonal matrix D_1.

$$HH_{W3} = U * D_1 * V$$

Step 6. Get the watermark barcode by inverse wavelet transform.

2.3.2 The Watermark Extraction

The watermarking extraction algorithm is the inverse process of the embedding algorithm. Specific process is shown in Figure 4:

Extraction Specific steps are as follows:

Step 1. Do three layers of discrete wavelet transform on the watermark barcode to obtain the diagonal high-frequency coefficient wHH3.

Step 2. Do singular value decomposition on the high frequency coefficients to get the diagonal matrix D2.

Step 3. Use the inverse formula of watermark embedding formula to get the encrypted watermark information.

Step 4. Do chaotic decryption on the encrypted watermark information to obtain the watermarking image.

2.3.3 The Strength of Watermark Embedding

Because the maximum singular value determines the image quality, then we can make full use of properties of singular value to determine the embedding strength [14]:

1) The low-frequency approximation sub graphs have a large number of singular value coefficient;

2) The largest singular value is larger than the second largest singular value coefficient of the low-frequency approximation sub graphs;

3) The watermark image singular value coefficient is basically bigger than the singular value coefficient of the sub graph;

4) The maximum singular value coefficient of the three sub-bands sub graphs is small, and the modification of the maximum singular value coefficient can not be too large, not more than 1/2 of itself. Otherwise, it will lead to the occurrence of serious deformation of the watermarked image.

Based on the above basis, we can embed intensity a in the diagonal belt of QR code image, determined by the follows:

$$\alpha = \frac{1}{50}[\frac{\lambda_{max}^{w}}{\lambda_{max}^{HH_3}}] \quad (2)$$

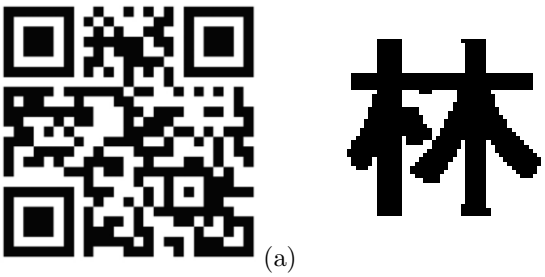
$\lambda_{max}^{HH_3}$ is the largest singular value coefficient of the diagonal belt after three layer DWT decomposition of the QR code image. λ_{max}^{w} is the largest singular value coefficient of watermark image. By the use of strength factor determined by Equation (2) to embed the watermark, the watermark QR code image can not only get the embedded watermark imperceptibility, but also has the very high PSNR.

Table 1: Contrast of the watermark PSNR and the QR code PSNR in different embedding factor

Embedded factor	Watermark PSNR	QR code PSNR	Embedded factor	Watermark PSNR	QR code PSNR
0.1	39.9825	55.7234	0.009	41.1290	55.7251
0.09	39.2029	55.7239	0.008	41.1650	55.7251
0.08	39.2029	55.7239	0.007	41.2002	55.7251
0.07	38.5651	55.7241	0.006	41.2654	55.7251
0.06	37.8698	55.7243	0.005	40.2329	55.7252
0.05	37.0222	55.7245	0.003	40.1655	55.7252
0.04	37.8219	55.7247	0.002	40.7303	55.7252
0.03	40.1965	55.7248	0.001	40.3763	55.7252
0.02	40.6210	55.7250	0.0005	40.3944	55.7252
0.01	41.0931	55.7251	0.0001v	41.3533	55.7252

3 Experimental Results and Analysis

The experiment uses the Matlab 7.8 environment, the original image is the QR code image with 512×512 pixels, and the watermark image is a binary image with 64×64 pixels, as shown in Figure 5.



(a) (b)

Figure 5: The original QR code and watermark image: (a) The QR code image; (b) The watermark image

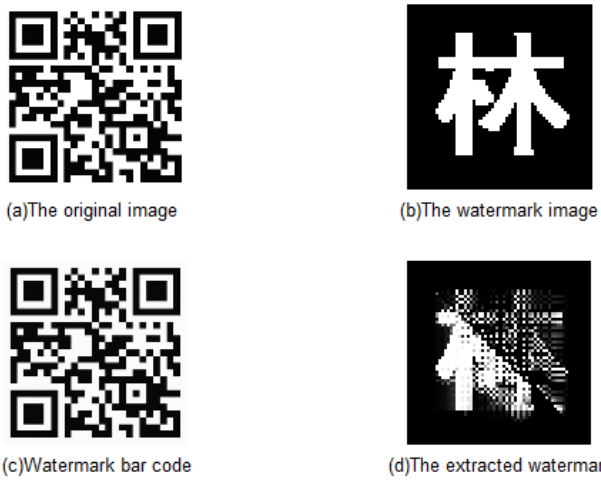


Figure 6: The watermarked image by DWT and the extracted watermark image

By running simulation program and selecting the QR code image and watermark image, we can get the experimental results shown in Figure 6. The results meet the watermark imperceptibility and it can be extracted correctly.

In order to ensure the robustness of the watermark, we must make the watermark embedding strength large enough, but not damage the visual quality of the image, so choosing a proper watermark embedding strength factor is the key to design the watermark barcode.

We use a number of different embedding factors for the watermark embedding in the experiments, for example, embedding factoris 0.1, 0.05, 0.01, 0.001, as shown in Figure 7. With the development of embedded factor reduction, the watermark experiments extracted gets clearer, and the water marked image is not significantly affected by QR code. When the embedded factor is equal to 0.01, the effective of the reduction on the extraction of the watermark becomes small gradually.

According to different embedded factors, we calculated the peak signal to noise ratio of the watermark and the QR code. The experimental results show that, when embedded factor is 0.01, the watermark barcode is well formed, and the extracted watermark can be identified well. When the embedded factor is larger than 0.1, it is difficult to identify the extracted watermark image. In the range of 0.1 to 0.001, with decreasing the intensity of embedded factor, the peak signal to noise ratio of the extracted watermark first decreases and then increases gradually, the peak signal to noise ratio of the watermark bar code shows a linear growth, when the embedding factor reaches down to 0.01, the peak signal to noise ratio tends to be stable. The experimental results are in Table 1.

4 Conclusions

This paper proposes a watermarking algorithm of anti-fake figure based on DWT and QR code. This article uses chaos XOR algorithm for encryption of watermark to ensure the security of the watermark. The new water-

Figure 7: Watermark barcode with different embedding factor and extracted watermark: (a) Embedded factor 0.1 ; (b) Embedded factor 0.05; (c) Embedded factor 0.01; (d) Embedded factor 0.001

marking barcode is designed by doing three layer wavelet decomposition to the QR code image, and by combinating the chaos encryption and singular value decomposition. The simulation results show that the method works well on the watermark embedding and its extracted, can obtain the correct content from the watermarked QR image and can also satisfy the invisibility of watermarking. Next, we will extend image segmentation [17] and learning-based method [3, 4, 7] to QR code watermarking in the future.

Acknowledgments

This project is supported by the National Nature Science Foundation of China (No. 61202496, 61304208), Hunan Provincial Natural Science Foundation of China (No. 13JJ2031), Science and Technology Program of Hunan Province (No. 2014SK2025). The "12.5" education planning project of Hunan province (No. XJK013CXX014), Hunan province and decision-making consulting research (No. 2013BZZ54).

References

[1] Y. Chao, L. Liu, L. Xue, et al., "Information hiding algorithm based on PDF417 barcode," *Computer Engineering*, vol. 36, no. 9, pp. 131–133, 2010.

[2] Denso Wave, The characteristics of QR code [EB/OL] 2013.

[3] B. Gu, V. S. Sheng, K. Y. Tay, "Walter Romano, and Shuo Li, Incremental Support Vector Learning for Ordinal Regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26. no. 7, pp. 1403–1416, 2015.

[4] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, S. Li, "Incremental learning for ν-Support Vector Regression," *Neural Networks*, vol. 67, pp. 140–150, July 2015.

[5] X. He, A. Hu, W. Zhang, et al., "QR barcode digital watermarking based on improved LSB algorithm," *Computerand Information Technology,* pp. 1–4, 2010.

[6] L. Liu, Y. Zhou, B. Zhang, "The QR code digital watermarking algorithm based on DCT and SVD," *Infrared and Laser Engineering*, vol. 42, no. S2, pp. 304–311, 2012.

[7] J. Li, X. Li, B. Yang, X. Sun, "Segmentation-based Image Copy-move Forgery Detection Scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[8] J. Liu, X. Li, "Fuzzy Chaotic watermarking algorithm based on wavelet domain," *Computer Engineering*, vol. 37, no. 8, pp. 132–134, 2011.

[9] G. Prabakaran, R. Bhavani, M. Ramesh, "A Robust QR code video watermarking scheme based on SVD and DWT composite domain," in *International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pp. 21–22, 2013.

[10] S. Rungraungsilp, M. Ketcham, P. Surakote, and S. Vongpradhip, "Data hiding method for QR code based on watermark by comparing DCT with DWT domain," in *International Conference on Computer and Communication Technologies*, pp. 26–27, May 2012.

[11] V. Seenivasagam, R. Velumani, "A QR code based zero-watermarking scheme for authentication of medical images in teleradiology cloud," *Computational and Mathematical Methods in Medicine*, vol. 2013, no. 2013, pp. 1–16, 2013.

[12] The global mobile phonesecurity report in the first quarter of 2014: Chinese rank stop in the mobile phone virus infection [EB/OL], 2014.

[13] S. Vongpradhip and S. Rungraungsilp, "QR code using invisible watermarking in frequency domain," In *IEEE 9th International Conference on ICT and Knowledge Engineering*, pp. 47–52, 2012.

[14] R. Xie, H. Zhao,Y. Chen, "Anti-fake electronic ticket digital watermarking method based on QR codes," *Journal of Xiamen University*, vol. 52, no. 3, pp. 38–342, 2013.

[15] S. Xue, X. Chen, "Digital image watermarking algorithm based on chaotic encryption and SVD," *Computer Engineering*, vol. 38, no. 19, pp. 107–110, 2012.

[16] B. Zhu, "Study on digital watermarking algorithm for QR code based on LSB," *Journal of Chengdu Information and Technology University*, vol. 27, no. 6, pp. 541–546, 2012.

[17] Y. Zheng, B. Jeon, D. Xu et al., "Image segmentation by generalized hierarchical fuzzy C-means algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 2, pp. 961–973, 2015.

Jiaohua Qin received her BS in mathematics from Hunan University of Science and Technology, China, in 1996, MS in computer science and technology from National University of Defense Technology, China, in 2001, and PhD in computing science from Hunan University, China, in 2009. She is a professor at College of Computer Science and Information Technology, Central South University of Forestry and Technology, China. Her research interests include network and information

security, image processing and pattern recognition.

Ruxin Sun received his BS in computer science and technology from Central South University of Forestry and Technology, China, in 2013. He is currently pursuing his MS in computer technology at College of Computer Science and Information Technology, Central South University of Forestry and Technology, China. His research interests include information security, image processing and pattern recognition.

Xuyu Xiang received his BS in mathematics from Hunan Normal University, China, in 1996, MS degree in computer science and technology from National University of Defense Technology, China, in 2003, and PhD in computing science from Hunan University, China, in 2010. He is a professor at Central South University of Forestry and Technology, China. His research interests include network and information security, image processing, and internet of things.

Hao Li received his BS in computer science and technology from Zhengzhou University, China, in 2015. He is currently pursuing his MS in computer application technology at College of Computer Science and Information Technology, Central South University of Forestry and Technology, China. His research interests include information security and image processing.

Huajun Huang is currently a faculty member in the college of Computer and Information Engineering at Central South University of Forestry & Technology. His overall research area include of webpage information hiding and hidden information detection, XML watermarking, anti-phishing, mobile device forensics. Dr. Huang received his Ph.D. from Hunan University in 2007, M.S. degrees from Hunan University in Software Engineering (2004), and a B.A. in Applied Physics from Yunnan University (2001).

A Lightweight Generic Compiler for Authenticated Key Exchange from Non-interactive Key Exchange with Auxiliary Input

Zheng Yang[1], Chao Liu[1], Wanping Liu[1], Song Luo[1], Hua Long[1] and Shuangqing Li[2]
(Corresponding author: Hua Long)
School of Computer Science and Engineering, Chongqing University of Technology[1]
Chongqing 400054, China
College of Computer Science, Chongqing University[2]
Chongqing 400044, China
(Email: cqlongman@163.com)
(Received Sept. 3, 2015; revised and accepted Dec. 7 & Dec. 15, 2015)

Abstract

We introduce a new lightweight generic compiler that is able to transform any passively forward secure two-message key exchange (KE) protocols into authenticated key exchange (AKE) protocols with security in the presence of active adversaries who can reveal critical session specific information such as long-term or ephemeral secrets and can establish malicious parties. The compiler is built based on a new security notion regarding non-interactive key exchange with auxiliary input (NIKEA). The NIKEA is able to provide two security properties on the confidentiality and the unforgeability of shared key. Our new compiler is a very useful tool for the design of new AKE protocols in a modular and efficient way, that is suitable for resources constrained devices.

Keywords: Authenticated key exchange, non-interactive key exchange, protocol compiler, standard model

1 Introduction

Authenticated key exchange (AKE) is a cryptographic primitive which enables two parties to compute a session key with an assurance that the generated key is only known by these intended communication partners. In many application systems, AKE protocols usually serve as an important building block to protect the communication data over insure networks.

AKE Compilers. It is known to be a generic security strengthening transformation that pushes forward the modular design of AKE protocols. An interesting fashion of AKE compiler is to securely combine authentication protocols (AP) with passively secure key exchange protocols (KE) to yield AKE protocols that is referred as AP&KE style compiler [14] in the sequel, see the works in [14, 16, 18]. In this paper, we focus on a variant of this style AKE compiler where the implicit key authentication is guaranteed (instead of the explicit mutual authentication in previous works). Several advantages of AKE compilers are worth highlighting. First of all one could realize a AKE protocol with a rich collection of existing authentication and key exchange protocols which are specifically fit to a certain application scenario. On the second, a generic compiler would be very useful to avoid any modifications (which are often costly or error-prone in practice) in existing implementations of the input sub-protocols. To the last but not least, it could simplify the security analysis of the entire system, where the security of any resulting AKE protocol is directly inherited from the security proof of the AKE compiler.

While reviewing existing AKE compilers[14, 16, 18], we notice that they might be not efficient enough. Katz and Yung presented a generic compiler (which is referred to as KY compiler) for building group authenticated key exchange [16] based on passively secure group key exchange and digital signature. The KY compiler needs an additional communication round to the input protocol, in which each party chooses a random nonce and broadcasts it to its communication partners. In 2010, Jager et al. [14] introduced the first compiler (called as JKSS compiler) which accounts only for a constant number of additional messages (which is independent of the KE protocol) to be exchanged. But this scheme requires the KE protocol to output the session key to the compiler (unlike the KY compiler) and increases three additional communication

rounds in the compiler that might be not practical. Most recently, Li et al. [18] proposed three new AP&KE style compilers (which are referred to as LSYBS compilers). Unlike the KY and JKSS compilers, no nonce is required in the LSYBS compilers which instead rely on the entropy of the ephemeral keys of KE. As a result the LSYBS compilers are more round efficient than KY and JKSS compilers. However, we find out that all these compilers increase the communication rounds to the compiled KE protocols. Thus they might be not suitable for power constrained devices which need low latency of communication. In addition, the LSYBS compiler shows that if a KE protocol without long-term key is passively secure then each protocol message generated by the ephemeral generation function (EKGen) is unique. This uniqueness property is what enables the LSYBS to get rid of the random nonce used in previous compilers such as KY compiler and the compiler introduced by Jager et al. [14] (which will be referred to as JKSS compiler). However, their restriction on EKGen rules out a lot of key exchange protocol with long-term key. In this work we therefore try to broaden the range of KE that a AKE compiler can work on, i.e., without putting restriction on specific ephemeral key generation function.

Recently, Boyd et al. [4] and Cremers et al. [8] proposed two compilers respectively for two-message AKE protocols. However, these two compilers all aimed to compile (e)CK secure two-message protocols to achieve perfect forward secrecy without increasing protocol round and changing the internal execution of compiled protocols. However, they need very strong assumptions on the compiled protocols, i.e., they should be proved secure in the CK [5] model or the eCK model [17]. On the other hand, the (e)CK secure protocols without random oracles are inefficient. The computational costs of these two compilers are basically less computational efficient than above AP&KE style AKE compiler. But, to our best of knowledge, it is still an open question on how to build AP&KE style AKE compiler without increasing protocol round.

Non-interactive key exchange. Non-interactive key exchange (NIKE) is introduced to allow two parties to calculate a shared key based only on their long-term public keys without any interaction. NIKE has many real-world applications, e.g., establishing keys and enabling secure communications in mobile ad hoc and sensor networks where the energy cost of communication is prime concern [6, 10, 11]. The formal security of NIKE was studied by Freire et al. [10]. However, the limitation of NIKE is also obvious that it lacks of some important security properties of AKE, such as perfect forward secrecy or resilience of known key attacks. Once the long-term private key or the shared key of honest parties is leaked somehow then the security of the system cannot be guaranteed anymore. Hence we try to figure out a solution on key establishment to make a trade-off between round efficiency and AKE security properties.

Contributions. In this paper, we first present a new notion concerning non-interactive key exchange with auxiliary input (NIKEA). In contrast to ordinary NIKE [10], the share key of NIKEA is generated relying on long-term keys and an auxiliary input string aux (which could be for example timestamps, constants or other public information). Intuitively, the shared key generated with different aux would be distinct. Hence the leakage of some shared key may not affect the security of other shared key with distinct aux. This leads the confidential security property of NIKEA to be stronger than that of NIKE. Besides, the NIKEA has another interesting security property on unforgeability that adversary is unable to generated the shared key of uncorrupted honest parties with an auxiliary input aux that is not used by these parties before. A concrete NIKEA scheme is proposed, which is derived from the pairing-based NIKE scheme in [10]. Moreover we somehow optimize the algorithms to make it to be more efficient and practical. Namely we require the certificate authority to check the validity of registered public key rather than doing so in each execution of shared key generation. The new NIKEA scheme is proven secure without random oracles under standard assumptions.

On the next we introduce a new lightweight AP&KE style compiler that generically build secure AKE from secure NIKEA protocols and two-message passively forward secure two-message key exchange protocols. Namely we take the NIKEA as an authentication protocol. We observe that the forward secrecy property of KE would lead the message transcript of each session to be unique among its owner's sessions. One of the reasons that we choose NIKEA as our building block is that it can be efficiently realized. It is remarkable that the new compiler does not require any modifications in the underlying KE and underlying application based on such KE. It is thus easily applicable to existing systems what makes it to be very useful in real world applications. The main idea is to take the message outputted by each KE instance as the auxiliary input of NIKEA, where the generated shared key is used as one-time authentication token for such KE message. Unlike previous compilers [14, 16, 18], we do not increase any protocol round. All communication can be done within two moves. In addition, the generic compiler can also be efficiently instantiated for instance with concrete Diffie-Hellman based KE and NIKEA. Then the computational cost is approximately dominated by only three exponentiations. In a nutshell, the proposed compiler is suitable for resources constrained application environment (such as sensor networks). Furthermore, the security analysis of the compiler is given in the standard model, i.e., without assuming random oracles. The security result shows that our compiler satisfies well-known desirable security properties including resilience of chosen identity and public key attacks, known session key attacks and leakage of ephemeral secrets (from sessions non-related to test session), and provision of perfect forward secrecy. Although the resilience of key compromise impersonation attacks is not covered by our compiler, we

believe that it would still meet the security requirement in most applications.

Other Related Works. In our work an important AKE security property that we care about is the perfect forward secrecy. It is notorious that the PFS for TMAKE is non-trivial to achieve. 1n 2012, Cremers and Feltz [8] proposed a stronger security model (referred to as eCKw) to reformulate the wPFS notion based on a new concept so called *origin-session*. The resultant model is claimed to provide a slightly stronger form of wPFS than eCK model's. On the second, they further develop eCKw to model PFS that yields another new model (which is referred to as eCK-PFS). More interestingly, it is possible to transform any eCKw secure protocol (e.g. [23]) to be eCK-PFS secure using the signature based compiler in [8]. The implication relationship between eCK and eCKw models was studied in literature [8, 24]. In 2016, Yang and Zhang [25] introduced a new authenticated group key exchange (AGKE) model named g-eCK-PFS which particularly covers PFS. These above models (e.g. eCK-PFS and g-eCK-PFS) consider the security of AKE protocol in a very strong sense. This also leads the protocols being secure in these models to be inefficiency.

Some other GAKE protocols, for instance. [7, 9, 12, 19, 21, 22] have been recently proposed from different motivations. But the efficiency still needs to be optimized somehow. We stress that our construction idea can also be used in the group case to build efficient AGKE protocol. But the group NIKEA is required then.

2 Preliminaries and Definitions

In this section, we describe the cryptographic building blocks that will be used in the rest of Sections. The set of integers between 1 and n is denoted by $[n] = \{1, \ldots, n\}$. The notion $a \xleftarrow{\$} S$ denotes the action of sampling a uniformly random element a from a set S. Let '||' denote the operation concatenating two binary strings. Let $\mathcal{IDS}$ be an identity space.

2.1 Target Collision-Resistant Hash Functions

Let $\mathsf{TCRHF} : \mathcal{K}_{\mathsf{TCRHF}} \times \mathcal{M}_{\mathsf{TCRHF}} \to \mathcal{Y}_{\mathsf{TCRHF}}$ be a family of keyed-hash functions associated with key space $\mathcal{K}_{\mathsf{TCRHF}}$, message space $\mathcal{M}_{\mathsf{TCRHF}}$ and hash value space $\mathcal{Y}_{\mathsf{TCRHF}}$. The public key $hk_{\mathsf{TCRHF}} \in \mathcal{K}_{\mathsf{TCRHF}}$ of a hash function $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, \cdot)$ is generated by a PPT algorithm $\mathsf{TCRHF.KG}(1^\kappa)$ on input security parameter κ. If the hash key hk_{TCRHF} is obvious from the context, we write $\mathsf{TCRHF}(m)$ for $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, m)$.

Definition 1. TCRHF *is called* $(t_{\mathsf{TCRHF}}, \epsilon_{\mathsf{TCRHF}})$*-target-collision-resistant if for all* t_{TCRHF}*-time adversaries* $\mathcal{A}$ *it holds that*

$$\Pr\left[\begin{array}{c} hk_{\mathsf{TCRHF}} \xleftarrow{\$} \mathsf{TCRHF.KG}(1^\kappa), \\ m \xleftarrow{\$} \mathcal{M}_{\mathsf{TCRHF}}, \\ m' \leftarrow \mathcal{A}(1^\kappa, hk_{\mathsf{TCRHF}}, m), \\ m \neq m',\ m' \in \mathcal{M}_{\mathsf{TCRHF}}, \\ \mathsf{TCRHF}(m) = \mathsf{TCRHF}(m') \end{array}\right] \leq \epsilon_{\mathsf{TCRHF}},$$

where the probability is over the random bits of $\mathcal{A}$.

Normally target collision resistant functions can be realized with a specific cryptographic hash function such as MD5 and SHA.

2.2 Pseudo-Random Functions

Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ denote a family of deterministic functions, where $\mathcal{K}_{\mathsf{PRF}}$ is the key space, $\mathcal{D}_{\mathsf{PRF}}$ is the domain and $\mathcal{R}_{\mathsf{PRF}}$ is the range of PRF for security parameter κ. Let $\mathsf{RF} : \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ be a stateful uniform random function which takes as input a distinct message $x \in \mathcal{D}_{\mathsf{PRF}}$, and outputs a random element $y \xleftarrow{\$} \mathcal{R}_{\mathsf{PRF}}$. The input message x of RF and its output y is one-to-one map.

Definition 2. *We say that* PRF *is a* $(t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family, if it holds that* $|\Pr[\mathsf{EXP}^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\kappa) = 1] - 1/2| \leq \epsilon_{\mathsf{PRF}}$ *for all adversaries* $\mathcal{A}$ *that make a polynomial number of oracle queries* q *while running in time at most* t *in the following experiment:*

$\mathsf{EXP}^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\kappa)$	$\mathcal{F}(b, x)$
$b \xleftarrow{\$} \{0,1\}, k \xleftarrow{\$} \mathcal{K}_{\mathsf{PRF}}$;	*If* $x \notin \mathcal{D}_{\mathsf{PRF}}$ *then return* $\perp$*;*
$b' \leftarrow \mathcal{A}^{\mathcal{F}(b,\cdot)}(\kappa)$;	*If* $b = 1$ *then return* $\mathsf{PRF}(k, x)$;
If $b = b'$ *then return 1;*	*Otherwise return* $\mathsf{RF}(x)$;
Otherwise return 0;	

where $\epsilon_{\mathsf{PRF}} = \epsilon_{\mathsf{PRF}}(\kappa)$ *is a negligible function in the security parameter* κ*, and the number of allowed queries* q *is bound by* t.

2.3 The Bilinear Decision Diffie-Hellman Assumption

We first briefly recall some of the basic properties of symmetric bilinear groups. The bilinear groups will be parametrized by a symmetric pairing parameter generator, denoted by $\mathsf{PG.Gen}$. This is a polynomial time algorithm that on input a security parameter 1^κ, returns the description of two multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of the same prime order p, generators g for $\mathbb{G}$, and a bilinear computable pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The formal description of of the properties of such pairing operation can be found in [3], which is omitted here.

Let $\mathcal{PG} : (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$ denote the description of symmetric bilinear groups. The Bilinear Decisional Diffie-Hellman (BDDH) problem [15] is stated as follows: given the tuple $(a, b, c, \gamma) \in \mathbb{Z}_p$ as input, and output yes if $e(g, g)^\gamma = e(g, g)^{abc}$ and no otherwise..

Definition 3. *We say that the* BDDH *problem relative to generator* PG.Gen *is* $(t, \epsilon_{\mathsf{BDDH}})$*-hard, if the probability bound* $|\Pr[\mathsf{EXP}^{bddh}_{\mathsf{PG.Gen},\mathcal{A}}(\kappa) = 1] - 1/2| \leq \epsilon_{\mathsf{BDDH}}$ *holds for all adversaries* $\mathcal{A}$ *running in probabilistic polynomial time* t *in the following experiment:*

$\mathsf{EXP}^{bddh}_{\mathsf{PG.Gen},\mathcal{A}}(\kappa)$
$\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$;
$(a, b, c, \gamma) \xleftarrow{\$} \mathbb{Z}_p^*$; $b \xleftarrow{\$} \{0, 1\}$;
if $b = 1$ $\Gamma \leftarrow e(g, g)^{abc}$, *otherwise* $\Gamma \leftarrow e(g, g)^\gamma$;
$b' \leftarrow \mathcal{A}(1^\kappa, \mathcal{PG}, g^a, g^b, g^c, \Gamma)$;
if $b = b'$ *then return 1, otherwise return 0;*

where $\epsilon_{\mathsf{BDDH}} = \epsilon_{\mathsf{BDDH}}(\kappa)$ *is a negligible function in the security parameter* κ.

2.4 Notations for Two-message KE

In a two-message AKE protocol (TMKE), each party may send a single 'message'. The key exchange procedure is done within two pass and a common shared session key is generated to be known only by session participants, which is shown in Figure 1.

A general TMKE protocol may consist of four polynomial time algorithms $(\mathsf{TMKE.ST}, \mathsf{TMKE.KG}, \mathsf{TMKE.MSG}, \mathsf{TMKE.SKG})$ with following semantics:

- $pms \leftarrow \mathsf{TMKE.ST}(1^\kappa)$: On input 1^κ, outputs pms, a set of system parameters.
- $(sk^{ke}_{\mathsf{ID}}, pk^{ke}_{\mathsf{ID}}) \xleftarrow{\$} \mathsf{TMKE.KG}(pms, \mathsf{ID})$: This algorithm takes as input system parameters pms and a party's identity $\mathsf{ID} \in \mathcal{IDS}$, and outputs a pair of long-term private/public key $(sk^{ke}_{\mathsf{ID}}, pk^{ke}_{\mathsf{ID}}) \in \{\mathcal{PK}, \mathcal{SK}\}$.
- $m_{\mathsf{ID}_1} \xleftarrow{\$} \mathsf{TMKE.MSG}(pms, sk^{ke}_{\mathsf{ID}_1}, \mathsf{ID}_2, pk^{ke}_{\mathsf{ID}_2}, r_{\mathsf{ID}_1}, m_{\mathsf{ID}_2})$: This algorithm takes as input system parameters pms and the sender ID_1's secret key $sk^{ke}_{\mathsf{ID}_1}$, the intended receiver ID_2's public key $pk^{ke}_{\mathsf{ID}_2}$, a randomness $r_{\mathsf{ID}_1} \xleftarrow{\$} \mathcal{R}_{\mathsf{TMKE}}$ and a message $m_{\mathsf{ID}_2} \in \mathcal{M}_{\mathsf{TMKE}}$ from party ID_2, and outputs a message $m_{\mathsf{ID}_1} \in \mathcal{M}_{\mathsf{TMKE}}$ to be sent, where $\mathcal{R}_{\mathsf{TMKE}}$ is the randomness space and $\mathcal{M}_{\mathsf{TMKE}}$ is message space. We remark that the secret key $sk^{ke}_{\mathsf{ID}_1}$ of sender, the identity ID_2 and public key $pk^{ke}_{\mathsf{ID}_2}$ of receiver are only optional for generating the message.[1]
- $K \leftarrow \mathsf{TMKE.SKG}(pms, sk^{ke}_{\mathsf{ID}_1}, \mathsf{ID}_2, pk^{ke}_{\mathsf{ID}_2}, r_{\mathsf{ID}_1}, m_{\mathsf{ID}_2})$: This algorithm take as the input system parameters pms and ID_1's secret key sk_{ID_1}, a public key $pk^{ke}_{\mathsf{ID}_2}$ of ID_2, a randomness $r_{\mathsf{ID}_1} \xleftarrow{\$} \mathcal{R}_{\mathsf{TMKE}}$ and a received message m_{ID_2} from party ID_2, and outputs session key $K \in \mathcal{K}_{\mathsf{TMKE}}$, where $\mathcal{K}_{\mathsf{TMKE}}$ is the session key space.

We say that the TMKE.SKG algorithm is correct, if for all $(sk^{ke}_{\mathsf{ID}_1}, pk^{ke}_{\mathsf{ID}_1}) \xleftarrow{\$} \mathsf{TMKE.KG}(\mathsf{ID}_1)$ and $(sk^{ke}_{\mathsf{ID}_2}, pk^{ke}_{\mathsf{ID}_2}) \xleftarrow{\$} \mathsf{TMKE.KG}(\mathsf{ID}_2)$, for all $r_{\mathsf{ID}_1}, r_{\mathsf{ID}_2} \xleftarrow{\$} \mathcal{R}_{\mathsf{TMKE}}$ and for all messages $m_{\mathsf{ID}_1} \xleftarrow{\$} \mathsf{TMKE.MSG}(sk^{ke}_{\mathsf{ID}_1}, \mathsf{ID}_2, pk^{ke}_{\mathsf{ID}_2}, r_{\mathsf{ID}_1}, \emptyset)$ and $m_{\mathsf{ID}_2} \xleftarrow{\$} \mathsf{TMKE.MSG}(sk^{ke}_{\mathsf{ID}_2}, \mathsf{ID}_1, pk^{ke}_{\mathsf{ID}_1}, r_{\mathsf{ID}_2}, m_{\mathsf{ID}_1})$, it holds that

$$\mathsf{TMKE.SKG}(sk_{\mathsf{ID}_1}, \mathsf{ID}_2, pk_{\mathsf{ID}_2}, r_{\mathsf{ID}_1}, m_{\mathsf{ID}_2}) = \mathsf{TMKE.SKG}(sk_{\mathsf{ID}_2}, \mathsf{ID}_1, pk_{\mathsf{ID}_1}, r_{\mathsf{ID}_2}, m_{\mathsf{ID}_1})$$

A the system initiation phase, the parameters would be generated as $pms \leftarrow \mathsf{TMKE.ST}(1^\kappa)$, where pms might be ignored in the description of other algorithms of TMKE for simplicity. The Figure 1 briefly illustrates the generic protocol execution of TMKE on input pms.

Please note that if in the above execution, if the party ID_2's message m_{ID_2} is generated to be independent of m_{ID_1} then the TMKE is a one-round AKE protocol, i.e. $m_{\mathsf{ID}_2} \xleftarrow{\$} \mathsf{TMKE.MSG}(sk^{ke}_{\mathsf{ID}_2}, \mathsf{ID}_1, pk^{ke}_{\mathsf{ID}_1}, r_{\mathsf{ID}_2}, \emptyset)$. The independence property of one-round AKE enables parties to run protocol instances simultaneously (which is a key feature of one-round protocol).

3 Non-Interactive Key Exchange with Auxiliary Input

In the subsection, we introduce a new security notion regarding non-interactive key exchange with auxiliary input (NIKEA).

3.1 Notions for Non-Interactive Key Exchange with Auxiliary Input

We consider a NIKEA scheme in the public key setting consists of three algorithms: NIKEA.Setup, NIKEA.KG and NIKEA.ShareKey associated with an identity space $\mathcal{IDS}$ and a shared key space $\mathcal{K}_{\mathsf{NIKEA}}$, in which those algorithms are defined as follows:

- $pms^{nikea} \leftarrow \mathsf{NIKEA.Setup}(1^\kappa)$: This algorithm takes as input a security parameter κ and outputs a set of system parameters pms^{nikea}. The parameters pms^{nikea} might be implicitly used by other algorithms for simplicity.
- $(sk_{\mathsf{ID}}, pk_{\mathsf{ID}}, \mathsf{pf}_{\mathsf{ID}}) \xleftarrow{\$} \mathsf{NIKEA.KG}(\mathsf{ID})$: This algorithm takes as input an identity ID, and outputs a pair of long-term secret/public key $(sk_{\mathsf{ID}}, pk_{\mathsf{ID}})$ and corresponding proof $\mathsf{pf}_{\mathsf{ID}}$ for key registration.
- $K \leftarrow \mathsf{NIKEA.ShareKey}(\mathsf{ID}_1, sk_{\mathsf{ID}_1}, \mathsf{ID}_2, pk_{\mathsf{ID}_2}, aux)$: This algorithm takes as input an identity ID_1, a secret key sk_{ID_1} along with another identity ID_2 and corresponding public key pk_{ID_2}, and an auxiliary input string $aux \in \{0, 1\}^*$, and outputs either a shared key $K \in \mathcal{K}_{\mathsf{NIKEA}}$ for the two parties, or a failure symbol $\perp$. This algorithm is assumed to always output $\perp$ if the input identities are not distinct.

[1] Please note that if ID_1 is initiator then $m_{\mathsf{ID}_2} = \emptyset$.

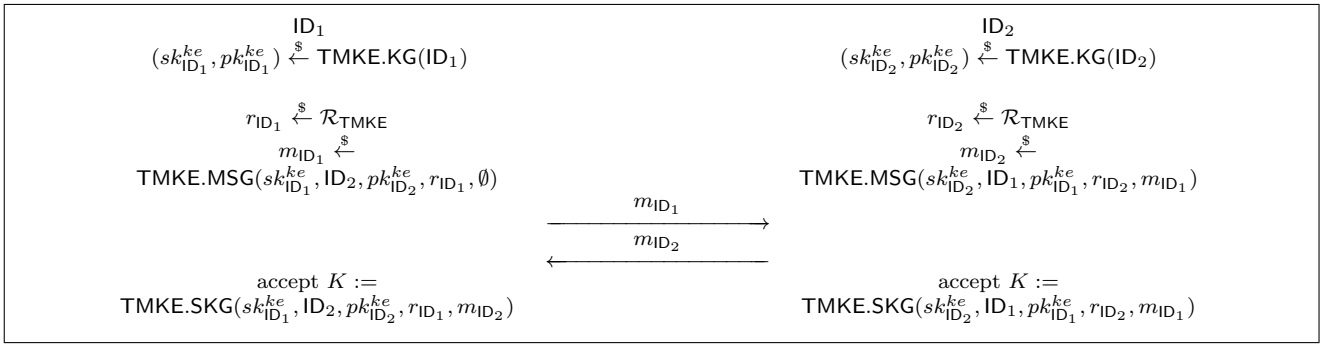


Figure 1: General TMKE protocol

For correctness, we require that, for a tuple of identities $(\mathsf{ID}_1, \mathsf{ID}_2)$, and corresponding key pairs $(sk_{\mathsf{ID}_1}, pk_{\mathsf{ID}_1})$ and $(sk_{\mathsf{ID}_2}, pk_{\mathsf{ID}_2})$ and the same aux, the algorithm $\mathsf{NIKEA.ShareKey}$ should satisfy the constraint:

– $\mathsf{NIKEA.ShareKey}(\mathsf{ID}_1, sk_{\mathsf{ID}_1}, \mathsf{ID}_2, pk_{\mathsf{ID}_2}, aux) =$ $\mathsf{NIKEA.ShareKey}(\mathsf{ID}_2, sk_{\mathsf{ID}_2}, \mathsf{ID}_1, pk_{\mathsf{ID}_1}, aux)$

One of the differences between the notion of NIKE in [10] and the above notion of NIKEA is that the $\mathsf{NIKEA.ShareKey}$ algorithm (in the later notion) requires an additional input aux for shared key generation. The NIKE in [10] can be seen as a special NIKEA with empty $aux = \emptyset$. In order to run the NIKEA correctly, two parties should share the aux somehow. For example the aux could be synchronized timestamps or constants or other public information.[2] In contrast to the NIKE, the NIKEA might be useful to generate either one-time shard key or authentication token for aux (see our upcoming AKE proposal). In a nutshell, the NIKEA can provide more functions than NIKE. In the following, we formally describe the security notion of NIKEA.

3.2 Security Definition for NIKEA

We describe the formal security model for two party PKI-based NIKEA protocols, that is modified from the CKS-like model [10] for NIKE. Besides we do slightly modification on modelling public key registration. Specifically, each party ID_i might be required to provide extra information (denoted by $\mathsf{pf}_{\mathsf{ID}_i}$) to prove that the registered public key is sound. Let $\{\mathsf{Honest}, \mathsf{Dishonest}\}$ be two vector lists. In order to formulate the capabilities of active adversaries against NIKEA, the adversaries are allowed to ask the following queries:

- $\mathsf{RegisterHonest}(\mathsf{ID})$: On input an identity $\mathsf{ID} \in \mathcal{IDS}$, if $\mathsf{ID} \notin \{\mathsf{Honest}, \mathsf{Dishonest}\}$ then $\mathcal{C}$ runs $\mathsf{NIKEA.KG}(pms^{nikea}, \mathsf{ID})$ to generate a long-term secret/public key pair $(sk_{\mathsf{ID}}, pk_{\mathsf{ID}}) \in (\mathcal{PK}, \mathcal{SK})$ and adds the tuple $(\mathsf{ID}, sk_{\mathsf{ID}}, pk_{\mathsf{ID}})$ into the list Honest, and returns pk to $\mathcal{A}$; as otherwise a failure symbol $\perp$ is returned. This query is allowed to be asked at most twice. Parties established by this query are called honest.

- $\mathsf{EstablishParty}(\mathsf{ID}_\tau, pk_{\mathsf{ID}_\tau}, \mathsf{pf}_{\mathsf{ID}_\tau})$: This query allows the adversary to register an identity ID_τ and a long-term public key pk_{ID_τ} on behalf of a party ID_τ, if the $\mathsf{ID}_\tau \notin \{\mathsf{Honest}, \mathsf{Dishonest}\}$ and pk_{ID_τ} is ensured to be sound by evaluating the non-interactive proof $\mathsf{pf}_{\mathsf{ID}_\tau}$. We only require that the proof is non-interactive in order to keep the model simple. Parties established by this query are called dishonest.

- $\mathsf{RevealKey}^{nikea}(\mathsf{ID}_1, \mathsf{ID}_2, aux)$: On input a tuple of identities $(\mathsf{ID}_1, \mathsf{ID}_2)$, $\mathcal{C}$ returns a failure symbol $\perp$ if both parties ID_1 and ID_2 are dishonest. Otherwise $\mathcal{C}$ runs $\mathsf{NIKEA.ShareKey}$ using the secret key of one of the honest parties in $(\mathsf{ID}_1, \mathsf{ID}_2)$ and the public key of the other party and the aux given by adversary, and returns the result to $\mathcal{A}$.

- $\mathsf{Test}^{nikea}(\mathsf{ID}_1, \mathsf{ID}_2, aux)$: Given two identities $(\mathsf{ID}_1, \mathsf{ID}_2)$ and string aux, the challenger $\mathcal{C}$ returns a failure symbol $\perp$ if one of following condition holds: (i) $\mathsf{ID}_1 = \mathsf{ID}_2$, (ii) $\mathsf{ID}_1 \notin \mathsf{Honest}$ or (iii) $\mathsf{ID}_2 \notin \mathsf{Honest}$. Otherwise the challenger $\mathcal{C}$ samples a random bit $b \xleftarrow{\$} \{0, 1\}$, and it answers this query in terms of the bit b. Specifically, if $b = 1$, $\mathcal{C}$ runs $\mathsf{NIKEA.ShareKey}$ using the secret key of ID_1 and the public key of ID_2 to obtain the shared key K_1; else if $b = 0$, the challenger generates a random key K_1. $\mathcal{C}$ returns K_b to adversary. This query can be queried only once.

Security Experiment for Confidentiality $\mathsf{EXP}^{\mathsf{NIKEA}, ind-cma}_{\mathsf{NIKEA}, \mathcal{A}}(\kappa)$: On input security parameter κ, the security experiment is proceeded as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ based on a non-interactive key exchange protocol with auxiliary input NIKEA, where the following steps are performed:

1) The $\mathcal{C}$ first runs $pms^{nikea} \leftarrow \mathsf{NIKEA.Setup}(1^\kappa)$ and gives pms^{nikea} to adversary $\mathcal{A}$.

[2] For example, the aux can be periodically distributed by certain trusted key management center.

2) The adversary $\mathcal{A}$ may interact with challenger $\mathcal{C}$ with $\mathsf{RegisterHonest}$, $\mathsf{EstablishParty}$, $\mathsf{RevealKey}^{nikea}$ queries as defined above.

3) Eventually, the adversary may terminate with outputting a bit b'.

4) At the end, the experiment returns 1 if all following conditions hold: (i) the adversary $\mathcal{A}$ has issued a Test^{nikea} query on input $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$ in either identity order, (ii) Both parties $\mathsf{ID}_1^*, \mathsf{ID}_2^* \in \mathsf{Honest}$, (iii) $\mathcal{A}$ has not issued $\mathsf{RevealKey}^{nikea}$ query with input $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$ in either identity order, and $b = b'$; Otherwise 0 is returned.

Definition 4. *A two party NIKEA protocol Σ is called $(t, \epsilon_{\mathsf{NIKEA\text{-}IND}})$-shared-key-secure if it holds that $|\Pr[\mathsf{EXP}^{\mathsf{NIKEA},ind-cma}_{\Sigma,\mathcal{A}}(\kappa) = 1] - 1/2| \leq \epsilon_{\mathsf{NIKEA\text{-}IND}}$ for all adversaries $\mathcal{A}$ running within time t in the above security experiment and for some negligible probability $\epsilon_{\mathsf{NIKEA\text{-}IND}} = \epsilon_{\mathsf{NIKEA\text{-}IND}}(\kappa)$ in the security parameter κ.*

The above security definition provide a stronger security guarantee than the CKS-light security [10], that allows the adversary to ask $\mathsf{RevealKey}^{nikea}$ queries to under attacked parties (as long as these queries have distinct inputs to Test^{nikea} query's).[3] In other words, the leaked shared key would not affect the shared key with distinct aux.

On the next we show another interesting security property of NIKEA, i.e., the unforgeability of the shared key associated with aux. Informally speaking the adversary is unable to output a shared key which is not generated by uncorrupted honest parties. A NIKE scheme combines with an authentication protocol (e.g., the one based on message authentication code) might fulfill the same security attribute as NIKEA. But this is not efficient and would require more security assumptions (comparing to using NIKEA).

Security Experiment for Unforgeability $\mathsf{EXP}^{\mathsf{NIKEA},euf-cma}_{\mathsf{NIKEA},\mathcal{A}}(\kappa)$: On input security parameter κ, the security experiment is proceeded as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ based on a non-interactive key exchange protocol with auxiliary input NIKEA, where the following steps are performed:

1) The $\mathcal{C}$ first run $pms^{nikea} \leftarrow \mathsf{NIKEA.Setup}(1^\kappa)$ and gives pms^{nikea} to adversary $\mathcal{A}$.

2) The adversary $\mathcal{A}$ may interact with challenger $\mathcal{C}$ with $\mathsf{RegisterHonest}$, $\mathsf{EstablishParty}$, $\mathsf{RevealKey}^{nikea}$ queries as defined above.

3) Eventually, the adversary may terminate with outputting a tuple $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*, K^*)$.

4) At the end, the experiment returns 1 if all following conditions hold: (i) both parties ID_1^* and ID_2^* are honest, (ii) $\mathcal{A}$ has not issued $\mathsf{RevealKey}^{nikea}$ query on input $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$ in either identity order, and (iii) $K^* = \mathsf{NIKEA.ShareKey}(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$; Otherwise 0 is returned.

Definition 5. *A two party NIKEA protocol Σ is called $(t, \epsilon_{\mathsf{NIKEA\text{-}EUF}})$-unforgeable-secure if it holds that $|\Pr[\mathsf{EXP}^{\mathsf{NIKEA},euf-cma}_{\Sigma,\mathcal{A}}(\kappa) = 1] - 1/2| \leq \epsilon_{\mathsf{NIKEA\text{-}EUF}}$ for all adversaries $\mathcal{A}$ running within time t in the above security experiment and for some negligible probability $\epsilon_{\mathsf{NIKEA\text{-}EUF}} = \epsilon_{\mathsf{NIKEA\text{-}EUF}}(\kappa)$ in the security parameter κ.*

Lemma 1. *Assume the NIKEA protocol Σ is $(t, \epsilon_{\mathsf{NIKEA\text{-}IND}})$-shared-key-secure, then it is also $(t, \epsilon_{\mathsf{NIKEA\text{-}EUF}})$-unforgeable-secure provided that $t \approx t'$ and $\epsilon_{\mathsf{NIKEA\text{-}EUF}} \leq \epsilon_{\mathsf{NIKEA\text{-}IND}}$.*

Proof. Suppose that there exists an adversary $\mathcal{A}_1$ which can win the unforgeability security experiment with output $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*, K^*)$ with overwhelming probability, then we could construct an adversary $\mathcal{A}_2$ using $\mathcal{A}_1$ to break the confidential property of Σ with the same advantage. Technically, $\mathcal{A}_2$ simulates the EUF-CMA security experiment for $\mathcal{A}_1$, and it forwards all queries from $\mathcal{A}_1$ to the challenger $\mathcal{C}$ in its own experiment and returns the corresponding answers to $\mathcal{A}_1$. Note that the triple $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$ was never queried by $\mathcal{A}_1$ to $\mathsf{RevealKey}^{nikea}$. When $\mathcal{A}_1$ outputs a $(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*, K^*)$, then $\mathcal{A}_2$ asks the $\mathsf{Test}^{nikea}(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$ with obtaining a test key K_b. If $K_b = K^*$ then $\mathcal{A}_2$ would know that the K_b is the real key with probability at least $\epsilon_{\mathsf{NIKEA\text{-}EUF}}$. □

3.3 A Concrete NIKEA Scheme

We here introduce a pairing-based NIKEA scheme which is derived from the pairing based NIKE in [10]. In this variant, the chameleon hash function used in [10] is replaced with a target collision resistant hash function TCRHF (to lower the assumption). In particular we make use of a pseudo-random function to not only generate the final shared key but also bind the identities and an auxiliary string to such shared key. This also enables us to deal with the chosen identity and public key attacks modelled by $\mathsf{EstablishParty}$ query. In addition, we require the trusted public key bulletin (such as Certificate Authority) to check the validity of registered public key rather than doing so in each $\mathsf{NIKEA.ShareKey}$ execution for efficiency consideration. This change would lead the NIKEA scheme to be more suitable to power constrained devices. It is noticeable only one exponentiation is required in the $\mathsf{NIKEA.ShareKey}$ algorithm of our modified scheme that is more efficient than the original one [10] which requires three pairings operations.

The concrete algorithms of our new NIKEA scheme between two parties ID_1 and ID_2 are defined as follows:

- $\mathsf{NIKEA.Setup}(1^\kappa)$. On input security parameter 1^κ, this algorithm is proceeded as the follows: (i) Run

[3] Note that if the query $\mathsf{EstablishParty}(\mathsf{ID}_\tau, pk_{\mathsf{ID}_\tau}, \mathsf{pf}_{\mathsf{ID}_\tau})$ is asked with $\mathsf{pf} = \emptyset$ and the query $\mathsf{RevealKey}^{nikea}(\mathsf{ID}_1, \mathsf{ID}_2, aux)$ is asked with $aux = \emptyset$, then the above model equals to the CKS-light model [10]. The number of $\mathsf{EstablishParty}$ queries is bound by the time t.

$\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$, and generate random values $u, u_0, u_1, u_2 \xleftarrow{\$} \mathbb{G}$; (ii) Run $hk_{\mathsf{TCRHF}} \xleftarrow{\$} \mathsf{TCRHF.KG}(1^\kappa)$; (iii) Return system parameters $pms^{nikea} := (hk_{\mathsf{TCRHF}}, u, u_0, u_1, u_2)$.

- $\mathsf{NIKEA.KG}(\mathsf{ID})$. On input a party's identity $\mathsf{ID} \in \mathcal{IDS}$, the key generation algorithm does the following steps: (i) Choose one random element $sk_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$ as its secret keys, and (ii) Compute corresponding public key $pk_{\mathsf{ID}} := e(u, g^{sk_{\mathsf{ID}}})$, and generate proof $\mathsf{pf} := (g^{sk_{\mathsf{ID}}}, (u_0 u_1^{h_{\mathsf{ID}}} u_2^{h_{\mathsf{ID}}^2})^{sk_{\mathsf{ID}}})$ where $h_{\mathsf{ID}} = \mathsf{TCRHF}(g^{sk_{\mathsf{ID}}})$. Then the public key is registered if $e(u_0 u_1^{h_{\mathsf{ID}}} u_2^{h_{\mathsf{ID}}^2}, g^{sk_{\mathsf{ID}}}) = e(u_0 u_1^{h_{\mathsf{ID}}} u_2^{h_{\mathsf{ID}}^2})^{sk_{\mathsf{ID}}}, g)$ and $e(g^{sk_{\mathsf{ID}}}, u) = pk_{\mathsf{ID}}$.
- $K \xleftarrow{\$} \mathsf{NIKEA.ShareKey}(\mathsf{ID}_1, sk_{\mathsf{ID}_1}, \mathsf{ID}_2, pk_{\mathsf{ID}_2}, aux)$. Given the private key sk_{ID_1} of party ID_1, party ID_2's public key pk_{ID_2} and auxiliary input string aux, the ID_1 generates the shared key $K := \mathsf{PRF}(pk_{\mathsf{ID}_2}^{sk_{\mathsf{ID}_1}}, \mathsf{ID}_1 || \mathsf{ID}_2 || aux)$.

Theorem 1. *Suppose the Bilinear Decisional Diffie-Hellman problem is $(t, \epsilon_{\mathsf{BDDH}})$-hard in $\mathcal{PG}$, the hash function TCRHF is $(t, \epsilon_{\mathsf{TCRHF}})$-target-collision-resistant and the pseudo-random function family is $(t, \epsilon_{\mathsf{PRF}})$-shared-key-secure as defined above. Then the proposed NIKEA scheme is $(t', \epsilon_{\mathsf{NIKEA\text{-}IND}})$-secure provided that $t \approx t'$ and $\epsilon_{\mathsf{NIKEA\text{-}IND}} \leq \epsilon_{\mathsf{TCRHF}} + \epsilon_{\mathsf{BDDH}} + \epsilon_{\mathsf{PRF}}$.*

The proof of this theorem is presented in Appendix A.

4 Security Model for Authenticated Key Exchange

In this section we present a security model for authenticated key exchange (AKE) that is extended from the model by Bellare and Rogaway [1] with additionally formulating the active attacks on chosen identity and public key attacks, known session key, leakage of ephemeral secret and perfect forward secrecy. In this model, the active adversary is provided with an 'execution environment' which emulates the real world execution of AKE protocols.

Execution Environment. In the execution environment, we fix a set of honest parties $\{\mathsf{ID}_1, \ldots, \mathsf{ID}_\ell\}$ for $\ell \in \mathbb{N}$, where ID_i ($i \in [\ell]$) is the identity of a party which is chosen uniquely from space $\mathcal{IDS}$. Each identity is associated with a long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i}) \in (\mathcal{SK}, \mathcal{PK})$ for authentication. Each honest party ID_i can sequentially and concurrently execute the protocol multiple times with different intended partners, this is characterized by a collection of oracles $\{\pi_i^s : i \in [\ell], s \in [d]\}$ for $d \in \mathbb{N}$.[4] Oracle π_i^s behaves as party ID_i carrying out a process to execute the s-th protocol instance (session), which has access to the long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i})$ and to all other public keys. Moreover, we assume each oracle π_i^s maintains a list of independent internal state variables: (i)pid_i^s – storing the identities and public keys of session participants which are sorted lexicographically in terms of identity, including ID_i; (ii) Φ_i^s – denoting the decision $\Phi_i^s \in \{\mathtt{accept}, \mathtt{reject}\}$; (iii)$\rho_i^s$ – denoting the role $\rho_i^s \in \{Initiator(I), Responder(R)\}$; (iv) sT_i^s – recording the transcript of messages sent by oracle π_i^s; (v) rT_j^t – recording the transcript of messages received by oracle π_i^s.

All those variables of each oracle are initialized with empty string which is denoted by the symbol $\emptyset$. At some point, each oracle π_i^s may complete the execution always with a decision state $\Phi_i^s \in \{\mathtt{accept}, \mathtt{reject}\}$.

Adversarial Model. An adversary $\mathcal{A}$ in our model is a PPT Turing Machine taking as input the security parameter 1^κ and the public information (e.g., generic description of above environment), which may interact with these oracles by issuing the following queries.

- $\mathsf{Execute}(ID_1, s_1, \mathsf{ID}_2, s_2)$: This query allows adversary to execute the protocol among unused oracles $\{\pi_i^{s_i}\}_{1 \leq i \leq 2}$, and responds with the transcript of the execution. The $\mathsf{pid}_i^{s_i}$ of each instance is set to $\{\mathsf{ID}_1, \mathsf{ID}_2\}$. We will write $\mathsf{Execute}(\mathsf{ID}_1, \mathsf{ID}_2)$ for short, where the identities are sorted lexicographically.
- $\mathsf{Send}(\mathsf{ID}_i, s, m)$: The adversary can use this query to send any message m of his own choice to oracle π_i^s. The oracle π_i^s will respond the next message m^* (if any) to be sent according to the protocol specification and its internal states. Oracle π_i^s would be initiated via sending the oracle the first message $m = (\top, \mathsf{pid}_i^s)$ consisting of a special initialization symbol $\top$ and a variable storing partner identities.
- $\mathsf{RevealKey}(\mathsf{ID}_i, s)$: Oracle π_i^s responds with the session key if $\Phi_i^s = \mathtt{accept}$.
- $\mathsf{RevealState}(\mathsf{ID}_i, s)$: Oracle π_i^s responds with randomness used to generate the session key of this oracle.
- $\mathsf{Corrupt}(\mathsf{ID}_i)$: Oracle π_i^1 responds with the long-term secret key sk_{ID_i} of party ID_i if $i \in [\ell]$; otherwise a failure symbol $\perp$ is returned.
- $\mathsf{EstablishParty}(\mathsf{ID}_\tau, pk_{\mathsf{ID}_\tau}, \mathsf{pf}_{\mathsf{ID}_\tau})$: This query allows the adversary to register an identity ID_τ ($\ell < \tau$ and $\tau \in \mathbb{N}$) and a static public key pk_{ID_τ} on behalf of a party ID_τ. Parties established by this query are called dishonest. This query is proceeded similarly to the one described in Section 3.2.
- $\mathsf{Test}(\mathsf{ID}_i, s)$: If the oracle has state $\Phi \neq \mathtt{accept}$, then the oracle π_i^s returns some failure symbol $\perp$. Otherwise it flips a fair coin b, samples a random element K_0 from key space $\mathcal{K}_{\mathsf{AKE}}$, and sets K_1 to the real session key of oracle π_i^s. Finally the key K_b is returned.

[4] An oracle in this paper might be alternatively written as $\pi_{\mathsf{ID}_i}^s$ which is conceptually equivalent to π_i^s.

Secure AKE Protocols. In order to denote the situation that two oracles are engaged in an on-line communication, we first define two notions regarding partnership, i.e. *matching sessions* (MS) and origin session (OS) [8], where the MS is used to formulate the security related to RevealKey query, and the OS is used to formulate the security related to RevealState and Corrupt queries.

Definition 6 (Matching sessions)**.** *We say that π_i^s has a* matching session *to π_j^t, if $\mathsf{pid}_i^s = \mathsf{pid}_j^t$, $\rho_i^s \neq \rho_j^t$, $rT_j^t = sT_i^s$ and $sT_j^t = rT_i^s$. The π_j^t is said to be the partner-oracle of π_i^s.*

Definition 7 (Origin session)**.** *We say that π_i^s has a* origin session *to π_j^t, if $rT_j^t = sT_i^s$. The π_i^s is said to be the origin-oracle of π_i^s.*

Correctness. We say an authenticated key exchange (AKE) protocol Π is correct, if two oracles π_i^s and π_j^t accept with matching sessions, then both oracles hold the same session key.

For the security definition, we need the notion of *freshness* of an oracle. in the sequel, we give two freshness definitions. Let π_i^s be an accepted oracle, π_j^t be an oracle (if it exists) having matching session to π_i^s, and π_l^v be an oracle (if it exists) having origin session to π_i^s.

Definition 8 (Passive Freshness)**.** *The oracle π_i^s is said to be* KE*-fresh if the following condition is held:*

- *$\mathcal{A}$ queried either* $\mathsf{RevealKey}(\pi_i^s)$ *or* $\mathsf{RevealKey}(\pi_j^t)$ *(if π_j^t exists).*

Definition 9 (Active Freshness)**.** *The oracle π_i^s is said to be* AKE*-fresh if none of the following conditions holds:*

1) *$\mathcal{A}$ queried* $\mathsf{EstablishParty}(\mathsf{ID}_j, pk_{\mathsf{ID}_j})$ *to some party $\mathsf{ID}_j \in \mathsf{pid}_i^s$.*
2) *$\mathcal{A}$ queried either* $\mathsf{RevealKey}(\pi_i^s)$ *or* $\mathsf{RevealKey}(\pi_j^t)$ *(if π_j^t exists).*
3) *$\mathcal{A}$ queried either* $\mathsf{Corrupt}(\mathsf{ID}_i)$ *or* $\mathsf{Corrupt}(\mathsf{ID}_j)$ *to some party $\mathsf{ID}_j \in \mathsf{pid}_i^s$.*
4) *$\mathcal{A}$ queried either* $\mathsf{RevealState}(\pi_i^s)$ *or* $\mathsf{RevealState}(\pi_l^v)$ *(if π_l^t exists).*

Let $M \in \{\mathsf{KE}, \mathsf{AKE}\}$ be a variable to denote two distinct security experiments.

Security Experiment $\mathsf{EXP}_{\Pi,\mathcal{A}}^M(\kappa)$: On input security parameter 1^κ, the security experiment is proceeded as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ based on (A)KE protocol Π, where the following steps are performed:

1) At the beginning of the game, the challenger $\mathcal{C}$ implements the collection of oracles $\{\pi_i^s : i \in [\ell], s \in [d]\}$, and generates ℓ long-term key pairs and corresponding proof $(pk_{\mathsf{ID}_i}, sk_{\mathsf{ID}_i}, \mathsf{pf}_{\mathsf{ID}_i})$ for all honest parties ID_i for $i \in [\ell]$ where the identity $\mathsf{ID}_i \in \mathcal{IDS}$ of each party is chosen uniquely. $\mathcal{C}$ gives adversary $\mathcal{A}$ all identities, public keys and proofs $\{(\mathsf{ID}_1, pk_{\mathsf{ID}_1}, \mathsf{pf}_{\mathsf{ID}_1}), \ldots, (\mathsf{ID}_\ell, pk_{\mathsf{ID}_\ell}, \mathsf{pf}_{\mathsf{ID}_\ell})\}$ as input.
2) If $M = \mathsf{KE}$, then $\mathcal{A}$ is allowed to ask a polynomial number of queries: Execute, Corrupt and RevealKey.
3) If $M = \mathsf{AKE}$, then $\mathcal{A}$ is allowed to ask a polynomial number of queries: Send, Execute, RevealState, Corrupt, EstablishParty and RevealKey.
4) At some point, $\mathcal{A}$ may issue a $\mathsf{Test}(\pi_i^s)$ query on an oracle π_i^s during the game with only once.
5) At the end of the game, the $\mathcal{A}$ may terminate with returning a bit b' as its guess for b of Test query.
6) Finally, 1 is returned if all following conditions hold:
   - $\mathcal{A}$ has issued a Test query to a M-fresh oracle π_i^s without failure,
   - $\mathcal{A}$ returned a bit b' which equals to b of Test-query;

   Otherwise 0 is returned.

Definition 10 (Session Key Security)**.** *We say that a correct key exchange protocol Π is (M, t, ϵ)-secure, if for any $\mathcal{A}$ runs the M security experiment within time t while having advantage $\epsilon = \epsilon(\kappa)$ in terms security parameter κ, it holds that*

- *If two oracles π_i^s and π_j^t accept with matching sessions, then except for ϵ the following conditions must be satisfied: (i) the oracle π_i^s has a unique matching session at party ID_j, and (ii) the oracle π_j^t has a unique matching session at party ID_i.*
- *If a* Test *query has been issued to a M-fresh oracle π_i^s, then the probability holds that $|\Pr[\mathsf{EXP}_{\Pi,\mathcal{A}}^M(\kappa) = 1] - 1/2| < \epsilon$.*

It is not hard to see that the KE security provides forward secrecy property in presence of passive adversary. Since we allow the adversary to Corrupt the long-term keys (if any) of principles.

5 Compiler for two-move AKE Protocol from NIKEA

In this section we propose a generic compiler that transforms a passively forward secure two-move TMKE protocol to a AKE protocol based on NIKEA. The resulting AKE protocol can provide the AKE security as modelled in Section 4 that covers a lot of well-known security attributes such as resilience of leakage ephemeral keys (from sessions not 'associated' with test session) and chosen identity and public key attacks (such as the unknown key share attacks or small sub-group attacks), and provision of perfect forward secrecy.

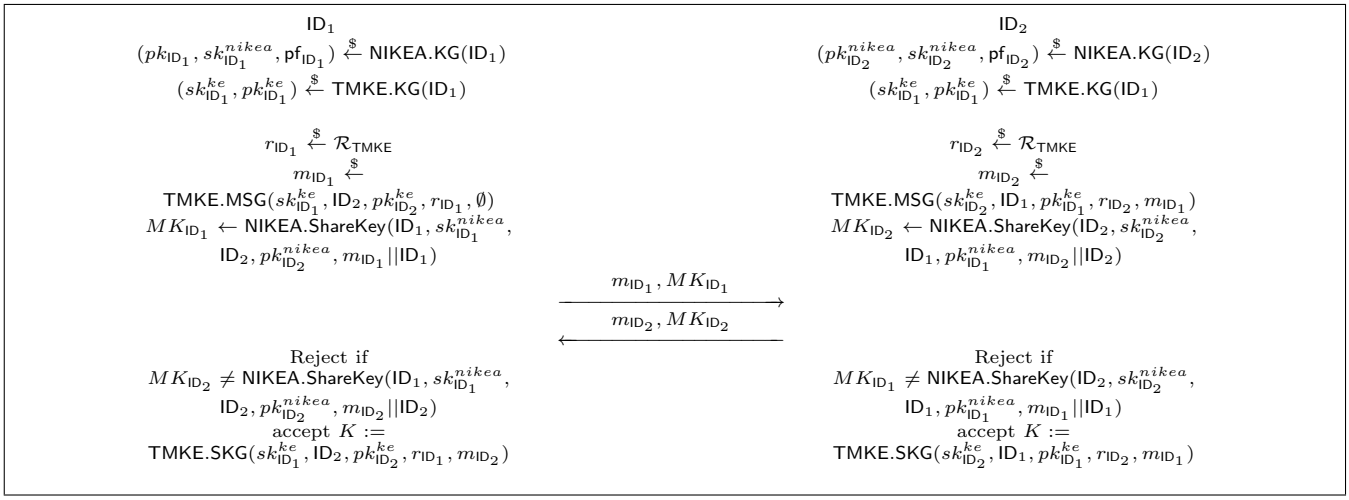


Figure 2: AKE protocol from NIKEA

Protocol Description. The compiler takes as input the following building blocks: (i)passively forward secure two-message key exchange protocol TMKE = ($\mathsf{TMKE.ST}$,$\mathsf{TMKE.KG}$,$\mathsf{TMKE.MSG}$,$\mathsf{TMKE.SKG}$), and CKS-like secure non-interactive key exchange scheme NIKEA = ($\mathsf{NIKEA.Setup}$, $\mathsf{NIKEA.KG}$, $\mathsf{NIKEA.ShareKey}$).

The parameters of the compiler consist of values generated by $pms^{nikea} \leftarrow \mathsf{NIKEA.Setup}(1^\kappa)$ and $pms^{ke} \leftarrow \mathsf{TMKE.ST}(1^\kappa)$. The generic compiler between two parties is shown as Figure 2.

Remark 1. In the above compiler, the NIKEA is used as a tool to authenticate the outgoing message of KE without changing it. Hence the KE and underlying application based on TMKE would have no 'awareness' on the increased compiler. Instead, security can be established by simply 'adding' the implementation of the compiler to the system. We stress that the computations on authentication tokens MK_{ID_1} and MK_{ID_2} can use the same shared key material (i.e., the $(pk_{\mathsf{ID}_2}^{nikea})^{sk_{\mathsf{ID}_1}^{nikea}}$). If one realizes the TMKE and NIKEA with Diffie-Hellman key exchange and the NIKEA presented in Section 3.2 respectively, then the overall computation cost would be approximately dominated by only three regular exponentiations (that is quite efficient). Moreover, the size of secret key is also very short that only one element in pairing group $\mathbb{G}$ is required. Such performance makes the resulting AKE protocol to be appealing to the resource constrained application environment (such as sensors networks).

The resilience of key compromise impersonation attacks is not covered by our compiler. In order to modify our compiler for achieving KCI resilience, a way is to use signature-based authentication protocol instead of NIKEA. But the computation cost will increase also. We leave out this as future work.

Comparisons. We summarize the comparisons between our proposal and some well known AKE compilers without random oracles in Table 1, i.e., the signature-based JKSS compiler [14] and the signature-based LSYBS compiler [18] which are referred to as JKSS_{SIG} and $\text{LSYBS}_{\text{SIG}}$ respectively. We instantiate the signature scheme in those compilers with the concrete one called $\text{Sig}_{\text{SRSA}}[\text{H}_{\text{cfs}}]$ [13] which is overall efficient on signing and verifying operations. Whereas the passive secure KE protocol in all compilers would be instantiated with the traditional Diffie-Hellman key exchange protocol.

Our comparisons are given from the following perspectives: (i) security assumptions; (ii) the number of exchanged messages sent by a party; (iii) overall computation cost of considered protocol; (iv) the communication round. Let 'DDH' denote the Decisional Diffie-Hellman assumption and 'SRSA' denote the strong RSA assumption. Let MAC denote the message authentication code. Let 'Exp' denote the regular exponentiation. In addition, we ignore the cost of PRF and MAC in the comparison.

Security Analysis. In the following, we are going to show that the new compiler is secure without appealing to random oracles.

Theorem 2. *Suppose the* TMKE *protocol is* $(\mathsf{TMKE}, t, \epsilon_{\mathsf{KE}})$*-secure and the NIKEA is* $(t, \epsilon_{\mathsf{NIKEA\text{-}EUF}})$*-unforgeable-secure. Then the proposed AKE compiler is* $(\mathsf{AKE}, t', \epsilon_{\mathsf{AKE}})$*-secure, provided that* $t \approx t'$ *and* $\epsilon_{\mathsf{AKE}} \leq d\ell \cdot \epsilon_{\mathsf{KE}} + d\ell^2 \cdot (\epsilon_{\mathsf{NIKEA\text{-}EUF}} + (d+1) \cdot \epsilon_{\mathsf{KE}})$.

We present the proof of this theorem in Appendix B.

6 Conclusions

We have presented a new security notion on non-interactive key exchange with auxiliary input (NIKEA). One of the advantages of NIKEA is that two parties may have a number of shared keys which are generated by different auxiliary input *aux*. We have also shown another interesting security property of NIKEA regarding unforgeability that adversary is unable to generate the

Table 1: Comparison

	Security Assumption	Message Length	Computation Cost	Communication Round
$JKSS_{SIG}$ [14]	SRSA, DDH, PRF, MAC	$9\mathbb{G}$	4 Exp	4
$LSYBS_{SIG}$ [18]	SRSA, DDH	$7\mathbb{G}$	4 Exp	2
Ours	BDDH, DDH, PRF	$2\mathbb{G}$	3 Exp	1

shared key of uncorrupted honest parties with an auxiliary input *aux* that is not used by these parties before. Based on such property, we have proposed a new lightweight AP&KE style compiler that generically build secure AKE from secure NIKEA protocols and passively secure two-move key exchange protocols without long-term keys. The new compiler is superior to previous similar works on perspectives of both communication and computation costs. Hence it is suitable for resources constrained application environment. As for a future work, it might be interesting to extend the idea of our compiler to group case for efficiency consideration, i.e. to build AGKE protocol from passively secure GKE and group NIKEA.

Acknowledgments

This study was supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ1500918, KJ1500904, KJ1401307), National Natural Science Foundation of China (Grant No. 11547148, 61503052), Research Project of Humanities and Social Sciences of Ministry of Education of China (Grant No. 15YJC790061), and Natural Science Foundation of Chongqing City (cstc2014jcyjA40024, cstc2013jcyjA40019).

References

[1] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptography (Crypto'93)*, LNCS 773, pp. 232–249, Springer, 1993.

[2] M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," in *Advances in Cryptography (Eurocrypt'06)*, LNCS 4004, pp. 409–426, Springer, 2006.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[4] C. Boyd and J. G. Nieto, "On forward secrecy in one-round key exchange," in *13th IMA International Conference Cryptography and Coding (IMACC'11*, LNCS 7089, pp. 451–468, Springer, 2011.

[5] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptography (Eurocrypt'01)*, LNCS 2045, pp. 453–474, Springer, 2001.

[6] Ç. Çapar, D. Goeckel, K. G. Paterson, E. A. Quaglia, D. Towsley and M. Zafer, "Signal-flow-based analysis of wireless security protocols," *Information and Computation*, vol. 226, pp. 37–56, 2013.

[7] T. Y. Chang, M. S. Hwang and W. P. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217–226, 2011.

[8] C. Cremers and M. Feltz, "Beyond eCK: Perfect forward secrecy under actor compromise and ephemeral-key reveal," *Designs, Codes and Cryptography*, vol. 74, no. 1, pp. 183–218, 2015.

[9] Z. Eslami, M. Noroozi and S. K. Rad, "Provably secure group key exchange protocol in the presence of dishonest insiders," *International Journal of Network Security*, vol. 18, no. 1, pp. 33–42, 2016.

[10] E. S. V. Freire, D. Hofheinz, E. Kiltz and K. G. Paterson, "Non-interactive key exchange," in *Public Key Cryptography*, pp. 254–271, 2013.

[11] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt and S. D. Wolthusen, "Strongly-resilient and non-interactive hierarchical key-agreement in manets," in *Computer Security (Esorics'08)*, pp. 49–65, Springer, 2008.

[12] D. He, C. Chen, M. Ma, S. Chan and J. Bu, "A secure and efficient password-authenticated group key exchange protocol for mobile ad hoc networks," *International Journal of Communication Systems*, vol. 26, no. 4, pp. 495–504, 2013.

[13] D. Hofheinz, T. Jager and E. Kiltz, "Short signatures from weaker assumptions," in *Advances in Cryptology (Asiacrypt'11)*, pp. 647–666, Springer, 2011.

[14] T. Jager, F. Kohlar, S. Schaege and J. Schwenk, "Generic compilers for authenticated key exchange," in *Advances in Cryptography (Asiacrypt'10)*, LNCS 6477, pp. 232–249, Springer, 2010.

[15] A. Joux, "A one round protocol for tripartite diffie–Hellman," in *Algorithmic Number Theory*, pp. 385–393, Springer, 2000.

[16] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *Journal of Cryptology*, vol. 20, no. 1, pp. 85–113, 2007.

[17] B. A. LaMacchia, K. Lauter and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, pp. 1–16, Springer, 2007.

[18] Y. Li, S. Schaege, Z. Yang, C. Bader and J. Schwenk, "New modular compilers for authenticated key exchange," in *International Conference on Applied*

Cryptography and Network Security, vol. 8479, pp. 1–18, Springer, 2014.

[19] Y. Li, D. Chen, W. Li, G. Wang and S. Paul, "A hybrid authenticated group key agreement protocol in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[20] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," Cryptology ePrint Archive, Report 2004/332, 2004. (`http://eprint.iacr.org/`)

[21] T. Y. Wu, Y. M. Tseng and T. T. Tsai, "A revocable id-based authenticated group key exchange protocol with resistant to malicious participants," *Computer Networks*, vol. 56, no. 12, pp. 2994–3006, 2012.

[22] C. Xu, H. Guo, Z. Li and Y. Mu, "New construction of affiliation-hiding authenticated group key agreement," *Security and Communication Networks*, vol. 6, no. 6, pp. 723–734, 2013.

[23] Z. Yang and W. Yang, "A practical strongly secure one-round authenticated key exchange protocol without random oracles," *Security and Communication Networks*, vol. 8, no. 6, pp. 1118–1131, 2015.

[24] Z. Yang, W. Yang, L. Zhu and D. Zhang, "Towards modelling perfect forward secrecy in two-message authenticated key exchange under ephemeral-key revelation," *Security and Communication Networks*, vol. 8, no. 18, pp. 3356–3371, 2015.

[25] Z. Yang and D. Zhang, "Towards modelling perfect forward secrecy for one-round group key exchange," *International Journal of Network Security*, vol. 18, no. 2, pp. 304–315, 2016.

Appendix A: Proof of Theorem 1

The proof will be given using a gamed based approach as in [2, 20]. Let break_δ be the event that the $\mathcal{A}$ correctly guesses the bit b sampled by the Test^{nikea} query in Game δ. Let $\mathsf{Adv}_\delta := \Pr[\mathsf{break}_\delta] - 1/2$ denote the advantage of $\mathcal{A}$ in Game δ.

Game 0 This is the original game with adversary $\mathcal{A}$. Thus we have that $\Pr[\mathsf{break}_0] - 1/2 = \epsilon_{\mathsf{NIKEA\text{-}IND}} = \mathsf{Adv}_0$.

Game 1 In this game we want to make sure that the received ephemeral keys are correctly formed. Technically the challenger proceeds exactly as before, but it aborts if there exist two distinct long-term public keys W and N such that $\mathsf{TCRHF}(W) = \mathsf{TCRHF}(V)$. According to the security property of underlying hash function, the above abortion event might occur with probability $\epsilon_{\mathsf{TCRHF}}$. Thus we have $\mathsf{Adv}_0 \leq \mathsf{Adv}_1 + \epsilon_{\mathsf{TCRHF}}$.

Game 2 In this game proceeds as the previous one, but we replace the key material $\beta^* = pk_{\mathsf{ID}_1^*}^{sk_{\mathsf{ID}_2^*}}$ of PRF with random value $\widetilde{\beta^*}$ where ID_1^* and ID_2^* are test parties. If there exists an adversary $\mathcal{A}$ which can distinguish the Game 2 from Game 1 then we can use it to construct an efficient distinguisher $\mathcal{D}$ to solve the BDDH problem as follows. Given a BDDH challenge instance $(\bar{g}, v, w, z, \Gamma)$, $\mathcal{D}$ sets $g := \bar{g}$, $u := z$, $pk_{\mathsf{ID}_1^*} := e(v, u)$ and $pk_{\mathsf{ID}_2^*} = e(w, u)$, and computes $h_{\mathsf{ID}_1} = \mathsf{TCRHF}(pk_{\mathsf{ID}_1^*})$ and $h_{\mathsf{ID}_2^*} = \mathsf{TCRHF}(pk_{\mathsf{ID}_2^*})$. Let $p(h) = p_0 + p_1 h + p_2 h^2 = (h - h_{\mathsf{ID}_1^*})(h - h_{\mathsf{ID}_2^*})$ be a polynomial of degree 2 over $\mathbb{Z}_p$ such that $p(h_{\mathsf{ID}_1^*}) = p(h_{\mathsf{ID}_2^*}) = 0$. Let $q(h) = q_0 + q_1 h + q_2 h^2$ be random polynomials of degree 2 over $\mathbb{Z}_p$. $\mathcal{D}$ next sets $u_0 = u^{p_0} g^{q_0}$, $u_1 = u^{p_1} g^{q_1}$ and $u_2 = u^{p_2} g^{q_2}$. $\mathcal{D}$ then answers the following queries:

- $\mathsf{RegisterHonest}(\hat{\mathsf{ID}})$: This query is simulated as the original one, except for the public keys for test parties ID_1^* and ID_2^* which are generated as above. In particular we have that $\mathsf{pf}_{\mathsf{ID}_1^*} = (v, v^{q(h_{\mathsf{ID}_1^*})})$ and $\mathsf{pf}_{\mathsf{ID}_2^*} = (w, w^{q(h_{\mathsf{ID}_2^*})})$, where $q(h_{\mathsf{ID}_1^*})$ and $q(h_{\mathsf{ID}_2^*})$ are known values. These are correct proofs for public keys of parties ID_1^* and ID_2^*, since $p(h_{\mathsf{ID}_1^*}) = p(h_{\mathsf{ID}_2^*}) = 0$.

- $\mathsf{EstablishParty}(\mathsf{ID}_\tau, pk_{\mathsf{ID}_\tau})$. Upon receiving a public key pk_{ID_τ} and an identity ID_τ from $\mathcal{A}$, the public key is registered if ID_τ has not been registered before and correspond proof $\mathsf{pf}_{\mathsf{ID}_\tau} = (\mathsf{pf}_{\mathsf{ID}_\tau,1}, \mathsf{pf}_{\mathsf{ID}_\tau,2})$ are evaluated correctly as protocol specification.

- $\mathsf{RevealKey}^{nikea}(\mathsf{ID}_1, \mathsf{ID}_2, aux)$. We assume this query is legitimate, otherwise $\mathcal{D}$ aborts. As for the case that there exists an honest user, say $\mathsf{ID}_2 \in \{\mathsf{ID}_1^*, \mathsf{ID}_2^*\}$ then $\mathcal{D}$ computes session key as
  $K = \mathsf{PRF}(e((\frac{\mathsf{pf}_{\mathsf{ID}_1,2}}{\mathsf{pf}_{\mathsf{ID}_1,1}^{q(h_{\mathsf{ID}_1})}})^{\frac{1}{p(h_{\mathsf{ID}_1})}}, pk_{\mathsf{ID}_2}), \mathsf{ID}_1 || \mathsf{ID}_2 || aux)$,
  where $h_{\mathsf{ID}_1} = \mathsf{TCRHF}(pk_{\mathsf{ID}_1})$.

- $\mathsf{Test}^{nikea}(\mathsf{ID}_1^*, \mathsf{ID}_2^*, aux^*)$: $\mathcal{D}$ returns $K^* = \mathsf{PRF}(\Gamma, \mathsf{ID}_1^* || \mathsf{ID}_2^* || aux^*)$.

This completes our simulation correctly. If $\Gamma = \mathsf{BDDH}(v, w, z)$, then the simulation is equivalent to Game 1; otherwise the simulation is equivalent to Game 2. At the end, $\mathcal{D}$ returns what $\mathcal{A}$ returns to BDDH challenger. If $\mathcal{A}$ can distinguish the real key from the random value, that implies $\mathcal{D}$ solves the BDDH problem. We therefore obtain that $\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + \epsilon_{\mathsf{BDDH}}$.

Game 3 In this game, the function $\mathsf{PRF}(\widetilde{\beta^*}, \cdot)$ computed in Test^{nikea} query is changed to a truly random function $\mathsf{RF}(\cdot)$. As the secret seed $\widetilde{\beta^*}$ is set to a truly random value due to previous game. If there exists an efficient adversary $\mathcal{A}$ who can distinguish the Game 3 from Game 2 with non-negligible advantage. Then we can construct an efficient algorithm $\mathcal{B}$ using $\mathcal{A}$ to break the security of PRF. In terms of the security of PRF, we have that $\mathsf{Adv}_2 \leq \mathsf{Adv}_3 + \epsilon_{\mathsf{PRF}}$. Note that in this game the session key returned by Test^{nikea} query is totally a truly random value which is independent to the bit b. Thus the probability that the adversary wins the game is $\mathsf{Adv}_3 = 0$.

Collect all the probabilities in above games, this theorem is proved.

Appendix B: Proof of Theorem 2

Let break_δ be the event that the $\mathcal{A}$ correctly guesses the bit b sampled by the Test-query in Game δ. Let $\mathsf{Adv}_\delta := \Pr[\mathsf{break}_\delta] - 1/2$ denote the advantage of $\mathcal{A}$ in Game δ. Let oracle $\pi_i^{s^*}$ denote the fresh test oracle and let $\pi_j^{t^*}$ denote the oracle having matching conversation to $\pi_i^{s^*}$.

Game 0 This is the original security game. We have that $\Pr[\mathsf{break}_0] - 1/2 = \epsilon_{\mathsf{AKE}} = \mathsf{Adv}_0$.

Game 1 This game proceeds exactly as the previous game but the challenger aborts if it fails to guess the test oracle $\pi_i^{s^*}$ and its intended communication partner ID_j. Since there are ℓ honest parties and d oracles for each party, the probability that the adversary guesses correctly is at least $1/(d\ell^2)$. Thus we have that $\mathsf{Adv}_0 \leq d\ell^2 \cdot \mathsf{Adv}_4$.

Game 2 In this game, the challenger proceeds exactly as previous game but it raises an abort event $\mathsf{abort}_{\mathsf{token}}$ that: the challenger aborts if test oracle accepts the incoming authentication token MK_{ID_j} which is not sent from its origin oracle. Due to the unforgeability of NIKEA, $\Pr[\mathsf{abort}_{\mathsf{token}}] \leq \epsilon_{\mathsf{NIKEA\text{-}EUF}}$. Thus we have that $\mathsf{Adv}_4 \leq \mathsf{Adv}_2 + \epsilon_{\mathsf{NIKEA\text{-}EUF}}$. In this game, we have that the test oracle must have origin oracle.

Game 3 In this game we are going to show that the test oracle has an origin oracle at each intended communication partner. The challenger proceeds as previous game but it aborts if either: (i) every (sub)-message (which could be, for instances, Diffie-Hellman key and identity) in the message transcript $m_i^{s^*}$ of test oracle $\pi_i^{s^*}$ has been sampled by some other oracle; or (ii) every (sub)-message in the message transcript $m_{\mathsf{ID}_j}^{t^*}$ of origin oracle $\pi_i^{s^*}$ (of test oracle) has been sampled by other oracle.

If the above abort event occurs in a non-negligible probability then there exists an adversary $\mathcal{B}$ which can break the KE security of P by running $\mathcal{A}$. The simulation of $\mathcal{B}$ is proceeded as follows:

- At the initiation phase, $\mathcal{B}$ first implements the collection of oracles $\{\pi_i^s : i \in [\ell], s \in [d]\}$. All long-term public/private key pairs for each honest user ID_i are generated and all public keys are given to adversary as input.
- Meantime, $\mathcal{B}$ generates the protocol messages for each oracle as protocol specification and answers all oracle queries honest except for the test oracle and its origin oracle.
- As for the test oracle and its partner oracles, $\mathcal{B}$ queries $\mathcal{C}_{\mathsf{KE}}$ for asking a $\mathsf{Execute}$ query to obtain T^*_{KE} and a Test query to obtain the session key $K^*_{b,\mathsf{KE}}$ of that $\mathsf{Execute}$ query. $\mathcal{B}$ simulates the test oracle chosen by $\mathcal{A}$ and its origin oracle using the transcript T^*_{KE} and $K^*_{b,\mathsf{KE}}$. $\mathcal{A}$ may keep asking oracle queries.
- $\mathcal{B}$ answers those oracle queries using secrets of her own choice. In particular the $\mathcal{B}$ would generate the signatures of oracles based on corresponding long-term key chosen by herself.

Assume that the adversary $\mathcal{A}$ leads two oracles $\pi_h^{v^*}$ (which is either test oracle or its origin oracle) and π_j^t to output the same message m^* without matching sessions. This means that the ephemeral secret key $esk_h^{v^*}$ (which equals to $esk_{\mathsf{ID}_j}^t$) used to generate the message m^* is known by $\mathcal{B}$. Since the oracle π_j^t is simulated by $\mathcal{B}$ honestly as protocol specification, i.e., esk_j^t is chosen by $\mathcal{B}$. Hence the $\mathcal{B}$ could break the KE security of P by computing the session key of test oracle using ephemeral secret $esk_h^{v^*}$. We therefore have that $\mathsf{Adv}_2 \leq \mathsf{Adv}_3 + \epsilon_{\mathsf{KE}}$.

This game also implies that each oracle π_i^s has a unique origin oracle. Hence the adversary $\mathcal{A}$ can not exploit $\mathsf{RevealKey}$ query to win the game.

Game 4 This game proceeds exactly as the previous game but the challenger aborts if it fails to guess the origin oracle $\pi_j^{t^*}$. Thus we have that $\mathsf{Adv}_3 \leq d \cdot \mathsf{Adv}_4$.

Game 5 Finally, we replace the key k^* of the test oracle $\pi_i^{s^*}$ and its partner oracle $\pi_j^{t^*}$ (if it exists) with the random value $\widetilde{k^*}$. Note that the KE protocol instance can be seen as being executed between the test oracle and its origin oracle due to Game 2. If there exists an adversary $\mathcal{A}$ which can distinguish this game from the previous game, then we use it to construct an algorithm $\mathcal{B}$ to break the passive security of key exchange protocol as follows. Assume that the adversary $\mathcal{B}$ interacts with the challenger $\mathcal{C}_{\mathsf{KE}}$. More specifically, $\mathcal{B}$ simulates the challenger in this game for $\mathcal{A}$ which is illustrated as follows:

- At the beginning, $\mathcal{B}$ implements the collection of oracles $\{\pi_i^s : i \in [\ell], s \in [d]\}$. All long-term public/private key pairs for each honest user ID_i are generated, and all public keys are given to adversary.
- Meantime, $\mathcal{B}$ generates the protocol messages for each oracle as protocol specification and answers all oracle queries honest except for the test oracle and its origin oracle.
- $\mathcal{B}$ queries a $\mathsf{Execute}$ query with obtaining message transcript T^* from $\mathcal{C}_{\mathsf{KE}}$, where one of the oracles involved in this query will be chosen as test oracle. $\mathcal{B}$ simulates the messages of test oracle and its origin oracle based on the transcript T^*. As for the $\mathsf{Test}(\pi_i^{s^*})$ query from $\mathcal{A}$, $\mathcal{B}$ answers it using the result of $\mathsf{Test}(\pi_i^{s^*})$ query returned by $\mathcal{C}_{\mathsf{KE}}$. If the test oracle $\pi_i^{s^*}$ has no matching session but has origin-oracle $\pi_j^{t^*}$ then the session key of $\pi_j^{t^*}$ can be computed using the ephemeral secret key (which is simulated by $\mathcal{B}$) of the origin-oracle of $\pi_j^{t^*}$. Please note that $\pi_j^{t^*}$ must also have origin-oracle due to the security of NIKEA.
- Eventually, $\mathcal{B}$ returns the bit b' from $\mathcal{A}$ to $\mathcal{C}_{\mathsf{KE}}$.

The simulation of $\mathcal{B}$ is perfect since $\mathcal{B}$ can always correctly answer all queries from $\mathcal{A}$. If $\mathcal{A}$ is able to correctly answer the bit b of Test-query with non-negligible probability, so does the adversary $\mathcal{B}$. Hence we obtain that $\mathsf{Adv}_4 \leq \mathsf{Adv}_5 + \epsilon_{\mathsf{KE}}$. In this game, the response to the Test query always consists of a random key, which is independent to the bit b flipped in the Test query. Thus we have $\mathsf{Adv}_5 = 0$. This theorem is proved by putting together of probabilities from above games.

Zheng Yang received the Master degree in Computer Science from Chongqing University in 2009. He got the doctor degree in IT-security from Ruhr-University Bochum, Germany in 2013. He is a researcher at Chongqing University of Technology, China. His main research interests include information security and cryptography.

Chao Liu received his B.S. degree and Ph.D. degree in Computer Science and Technology from Chongqing University in 2006 and 2013, respectively. He has been an instructor at School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His current research interests include impulsive systems, switched systems, neural networks, and network security.

Wanping Liu received the Ph.D. degree from the College of Computer Science, Chongqing University, China, in 2014. He is currently working in the School of Computer Science and Engineering, Chongqing University of Technology. His current research interests include discrete dynamical systems, mathematical modeling and network security.

Song Luo received the Ph.D. degree from the College of Computer Science, Peking University, China, in 2011. He is currently working in the School of Computer Science and Engineering, Chongqing University of Technology. His current research interests include Cryptology, and Network Security.

Hua Long received the Ph.D. degree from the College of Computer Science, Chongqing University, China, in 2010. He is currently working in the School of Computer Science and Engineering, Chongqing University of Technology. His current research interests include natural language processing, software engineering, pattern recognition, and network security.

Shuangqing Li received the Ph.D. degree from the College of Computer Science, Chongqing University, China, in 2010. He is currently working in the College of Computer Science, Chongqing University. His current research interests include software engineering, pattern recognition, cloud computing, and network security.

A High Payload Steganographic Scheme for Compressed Images with Hamming Code

Junlan Bai[1], Chin-Chen Chang[2]
(Corresponding author: Chin-Chen Chang)
School of Electronic Engineering, University of Electronic Science and Technology of China[1]
No. 2006 Xiyuan Ave, West Hi-Tech Zone, Chengdu, 611731, China
Department of Information Engineering and Computer Science, Feng Chia University[2]
No. 100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
(Email: alan3c@gmail.com)
(Received Aug. 24, 2015; revised and accepted Nov. 27, 2015 & Jan. 3, 2016)

Abstract

Data hiding schemes in the compressed domain have attracted more attention since the compressed image format is one of the most frequently transmitted formats over the Internet. Specifically, among various compression algorithms, Absolute Moment Block Truncation Coding (AMBTC) is a good choice due to its extremely low complexity and acceptable distortion. In this study, we propose a novel steganography using matrix embedding with Hamming code to insert secret message into AMBTC compressed bit-stream. Experimental results demonstrate that our scheme outperforms the other four existing BTC-based data hiding approaches in terms of embedding capacity, bit rate, and hiding efficiency.

Keywords: AMBTC compression, data hiding, hamming code

1 Introduction

With the rapid development of the contemporary society, new devices and powerful software make the world more digitalized and informationized. The wireless network offer ubiquitous channels to deliver and exchange various kinds of multimedia information, which becomes indispensable for humans' daily life. However, the potential brought about by freely uploading, downloading and manipulating data can't be fully realized without the protection of privacy. Data hiding technique [1, 14] was proposed to embed confidential data undetectably and imperceptibly in digital content.

Data hiding can be applied to three domains, which are spatial, frequency and compression domains of the cover images. Herein, the difference is the domain where the embedding happens. Compared with much research in the spatial domain and frequency domain data hiding schemes, data hiding techniques in the compression domain have not been fully explored. Currently, the compressed image format is frequently used transmission format over the Internet for reducing the storage space and transmission bandwidth. Since enormous digitalized data transmitted through the open channels occupies a relatively large part of bandwidth, compression techniques such as vector quantization (VQ) [10, 17, 18, 19, 20], JPEG [15], block truncation coding (BTC) [6], and Absolute Moment Block Truncation Coding (AMBTC) [11], are favored by many scholars. BTC was firstly proposed by Delp and Mitchell in 1979 [6]. It is a well-documented spatial image compression algorithm with low complexity compared to other compression techniques, such as JPEG [15]. BTC has been applied to graphics, digital video and colour digital imagery. Lema and Mitchell presented an AMBTC technique [11] to further increase the compression rate. In recent years, diverse data hiding schemes [2, 3, 4, 7, 8, 13, 21] based on the compression domain have been developed by academic researchers.

In 2002, Jo and Kim introduced a watermarking scheme [8] based on vector quantisation (VQ). It improved the degree of spreading watermark information by partitioning the codebook into three groups. An optimal vector quantizer codebook design contributes to increasing the steganographic image quality. However, it still offers worse visual quality of the stego image compared to hiding in BTC-compressed image. For the sake of improving the quality, Chuang and Chang [3] presented a scheme of embedding a majority of secret message into the smooth block of BTC-compressed image and the rest of secret data into the complex block in 2006. Although their technique achieved a relatively satisfactory quality, the hiding ability of each compressed block was less than one secret bit, which resulted in a rather low payload. In 2010, Chen et al. proposed an embedding algorithm [2] which exploited the relationship between the low- and high-means to judge whether to conceal confidential information into the corresponding compressed block or not. In [2], each

block can carry one bit or explore bits in the bitmap sometimes to embed extra secret data. In order to further enlarge the payload, Sun et al. presented a data hiding scheme [13] by losslessly embedding secret data in both the low- and high-mean tables in 2013. In 2015, Chang et al. proposed a data hiding scheme [4] for BTC compressed image by means of dynamic programming strategy. The dynamic programming strategy was employed to find a best bijective mapping relationship by using LSB replacement. In 2012, Kim et al. proposed a secret sharing scheme for hiding a watermark in two image shadows made from AMBTC based on Hamming Code [9]. Inspired by their method, we employ (7, 4) Hamming code to embed secret information in a digital image with higher embedding capacity than the existing data hiding schemes [2, 3, 4, 7, 8, 13, 21] based on the compression domain while remaining an acceptable visual quality.

The rest of the paper is organized as follows. Section 2 introduces the related techniques used in this paper. Our proposed scheme is presented in Section 3. The experimental results are illustrated in Section 4. Finally, the conclusions of the paper are stated in Section 5.

2 Preliminaries

This section will review the Absolute Moment Block Truncation Coding (AMBTC) algorithm, (7, 4) Hamming code and the matrix embedding.

2.1 Absolute Moment Block Truncation Coding (AMBTC)

In order to speed up the transmission process and save the bandwidth, Lema and Mitchell presented an AMBTC technique [11] for compressing grayscale and colour images in 1984. It outperforms BTC in computational complexity and mean squared error (MSE). AMBTC requires preserving two moments, mean and the first absolute central moment of image blocks, respectively. This algorithm starts off by partitioning the image into non-overlapping blocks with the size of $n \times n$. Denote $k = n \times n$ as the total number of pixels of the block. For each segmented block, the block mean value η is computed as follows:

$$\eta = \frac{1}{k}\sum_{i=1}^{k} x_i,$$

where x_i denotes the gray-level intensity of the pixel i in a block. Since AMBTC is a one-bit quantizer, η is utilized as a threshold for binarizing all the pixels in each block into two clusters with p and $k-p$ pixels, individually. If the intensity of the pixel is lower than η, it falls into the first cluster C_0. Otherwise, it falls into the second cluster C_1. The result is bitmap which is employed to record the distributions of the two quantization levels. The low- and high-mean used to represent a block is organized as follows:

$$a = \frac{1}{p}\sum_{x_i<\eta} x_i, \quad (1)$$

$$b = \frac{1}{k-p}\sum_{x_i\geq\eta} x_i, \quad (2)$$

where the two variables a and b are the quantization levels. Finally, the reconstructed AMBTC compression image is expressed as follows:

$$y_{i,j} = \begin{cases} a, & \text{if } h_{i,j}=0, \\ b, & \text{if } h_{i,j}=1, \end{cases} \quad h_{i,j} = \begin{cases} 0, & \text{if } x_{i,j}=C_0, \\ 1, & \text{if } x_{i,j}=C_1, \end{cases}$$

where $h_{i,j}$ is the element in the bitmap BM and $y_{i,j}$ denotes the pixel in the compressed AMBTC image. From the above equations, we can see each compressed block will generate a trio (a, b, BM), two quantization levels and one bitmap. An example of the AMBTC scheme is described in Figure 1. Denote that Figure 1(a) is the original image block containing 4×4 pixels. Firstly, the mean value η is computed as 138. Then the bitmap BM is achieved based on AMBTC algorithm and the result is illustrated in Figure 1(b). According to Equations (1) and (2), we can obtain the two quantization level values of a and b as 135 and 166, respectively. Finally, the generated compressed trio (135,166, 0011001000001101) is transmitted to the receiver. Herein, the receiver decodes this trio and reconstructs the image block shown in Figure 1(c).

123	104	158	192
137	119	188	129
133	101	121	130
141	168	115	149

(a) Original image block

0	0	1	1
0	0	1	0
0	0	0	0
1	1	0	1

(b) AMBTC bitmap

135	135	166	166
135	135	166	135
135	135	135	135
166	166	135	166

(c) Reconstructed image block

Figure 1: An example of AMBTC algorithm procedure

2.2 The (7, 4) Hamming Code

Hamming code [16] is widely used and favored by researchers due to its perfect structure and ability to detect and correct one error in a block of bits. It is a kind of block error correction linear codes. The (7, 4) Hamming code is used in this paper to operate bit modification. It encodes four original message bits $k = (c_6, c_5, c_4, c_3)$ by adding three extra parity check bits $r = (c_2, c_1, c_0)$ to form the transmitted codeword of length 7, which is ready for transmission.

In Hamming code, there are two important matrices: Generator matrix G and parity check matrix H, which are essential for encoding and decoding. At the encoding side, the codeword is obtained by multiplying the 4 data bits $k = (c_6, c_5, c_4, c_3)$ with the generator matrix

$G_{4\times7}$, and then applying modulo of 2. The resulting 7-bit codeword will be sent to the receiving side through a noisy channel. At the receiving side, the receiver reads the codeword, multiplies parity check matrix $H_{3\times7}$ with it and takes modulo of 2 again to check whether there is any error. The result is a syndrome vector of three bits. If the syndrome is '000', there is no error. If a single bit error occurs, the value is not equal to '000'. Then, we need to find which column of H is identical to the syndrome. For example, if the syndrome is '011' and the fourth column in H equals '011', flip the fourth bit of the received codeword and finally obtain the correct original data bits by ignoring the last three bits.

2.3 The Matrix Embedding with Hamming Code

Matrix embedding was originally introduced by Crandall [5] in 1998 to achieve high embedding efficiency, i.e., to reduce steganographic modifications to the cover images. The Hamming code is initially applied to matrix embedding, in which $n-k$ secret bits are inserted into n cover pixels by an $[n, k]$ Hamming code, of which the parity check matrix is H and $n = 2^k - 1$. Mao [12] designed a fast algorithm for matrix embedding in 2014. In the matrix embedding, an embedding group is defined as the cover vector $x = (x_1, x_2, ..., x_n)$ and the secret bits $m = (m_1, m_2, ..., m_{n-k})$. Change the positions of the columns in parity check matrix H to sort all the columns in ascending order in advance. Then, the mathematical model of embedding process is expressed as follows:

$$y = \mathrm{Emd}(x, m) = x + F(m - \mathrm{H}x),$$

where y is the received stego vector and $F(\cdot)$ is a function that transforms $(m - \mathrm{H}x)$ into a decimal value i and sets the i-th bit to '1' while other bits of the vector are set to '0'. At the receiving side, the receiver can easily extract the embedded message by:

$$\begin{aligned} m' &= \mathrm{Ext}(y) \\ &= \mathrm{H}(x + F(m - \mathrm{H}x)) \\ &= \mathrm{H}x + m - \mathrm{H}x = m, \end{aligned} \tag{3}$$

3 Proposed Scheme

In this section, the proposed scheme will be thoroughly presented. First, AMBTC algorithm is performed on the original grayscale cover image to obtain the compressed data that can be represented by a low-mean, a high mean, and a bitmap. Then, the secret message is concealed into the AMBTC compressed trio (a, b, BM). The advantage is to achieve a higher payload compared to other data hiding schemes performed on the compression domain. There are two phases for embedding in our proposed scheme. We firstly perform using the (7, 4) Hamming code on the low- and high-mean values of each compressed trio and exploit the relationship between them to hide one more secret bit. Then, we embed secret message into the bitmaps. The reason why we choose (7, 4) Hamming code is based on two considerations. On the one hand, we can embed three secret bits into seven cover bits. On the other hand, only one bit in the cover vector will be flipped after embedding, which means the modification to the cover vector is reduced to the minimum.

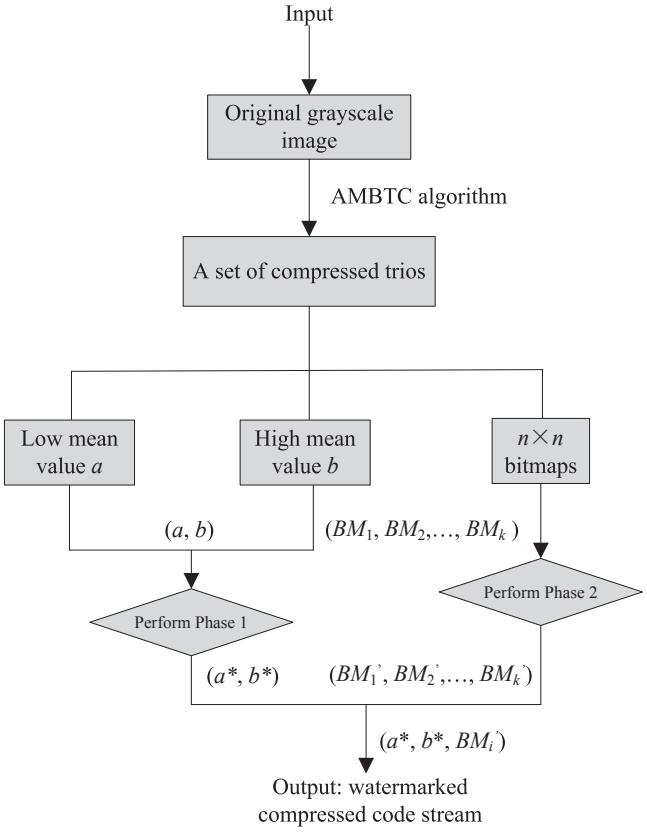


Figure 2: The flowchart of the embedding phase

3.1 Embedding Phase

After the original grayscale image is partitioned into a set of $n \times n$ blocks for AMBTC compression, each block undergoes the same process, thereby the description below towards a single block processing is adopted for simplicity. Normally, n equals 4. The flowchart of the embedding procedure which contains two phases is illustrated in Figure 2. A string of secret message is generated by a random number generator in advance.

The embedding phase 1: Embed secret message into two quantization levels.

Step 1. Extract four LSBs of low mean value a and three LSBs of high mean value b to constitute the column vector $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and read three secret message bits denoted as column vector $m = (m_1, m_2, m_3)$. Both of them are composed of the embedding group $\{x, m\}$.

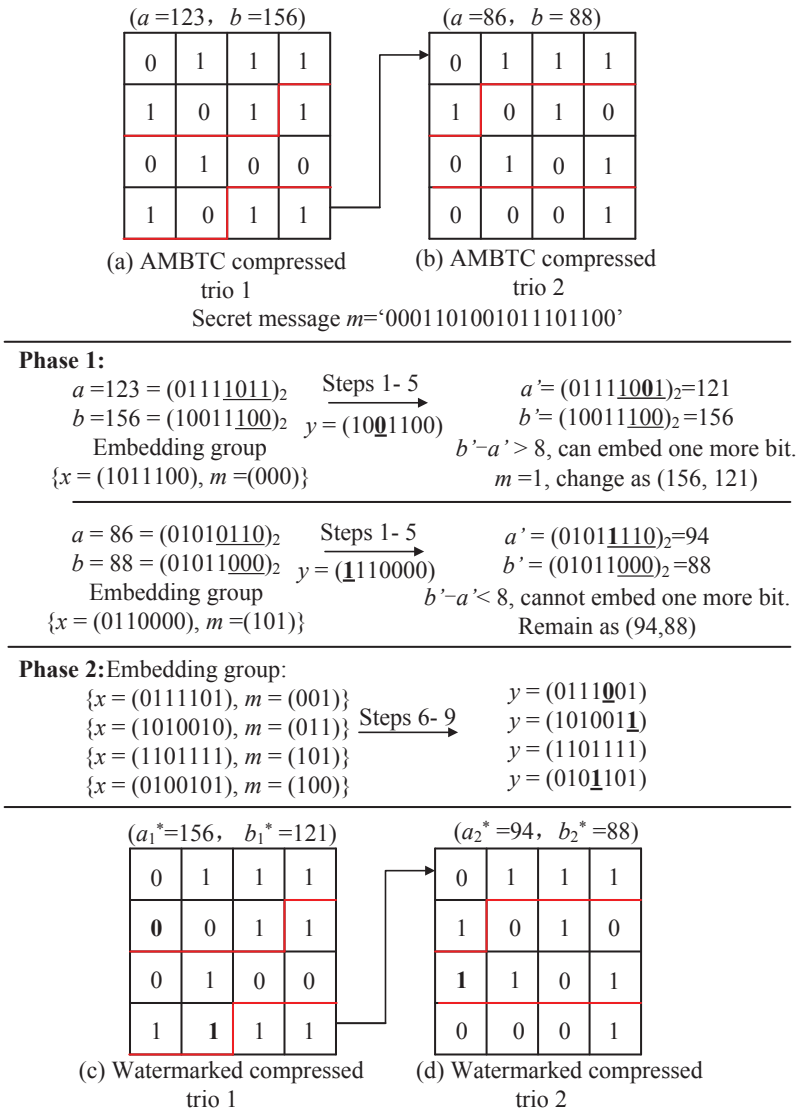


Figure 3: Example of embedding procedure of the proposed scheme

Step 2. Construct the parity check matrix H of (7, 4) Hamming code. Change the positions of the columns in H to ensure they array in ascending order in decimal form.

Step 3. Compute syndrome $s = m - \mathrm{H}x$, and transform s into a decimal number i. Change the i-th bit of vector x to obtain the stego vector $y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$. Each time, three secret bits are embedded into seven bits.

Step 4. Replace the four LSBs of low mean value a with (y_1, y_2, y_3, y_4) and three LSBs of high mean value b with (y_5, y_6, y_7) to obtain newly generated quantization pair (a, b).

Step 5. Compare the difference value between a' and b' to see whether they can be applied to the second embedding. If $b' - a' \leq 8$, this pair belongs to the unembeddable category. If not, we can embed one more secret bit through interchange operation. If the corresponding embedded bit is '1', exchange the two quantization levels, which becomes (b', a'). Otherwise, leave the location of (b', a') remain the same. Denote the final result of the quantization pair as (a^*, b^*).

It is notable that we set a threshold as eight to determine whether it is an embeddable group or not, because after operating Steps 1-4, the rangeability of a and b brought by matrix embedding may be eight and four respectively and only one of them will be changed. If a' is larger than b', it may suffer from confusion in the data extraction phase, because we cannot figure out whether the result is obtained by matrix embedding or interchange operation. In order to avoid ambiguity, we set a threshold to distinguish between these two cases. After all the quantization level pairs are processed, we conduct embedding in the bitmaps.

The embedding phase 2: Embed secret message into bitmaps.

Step 6. Concatenate each bitmap as (BM_1, BM_2,..., BM_k) to form a sequence of bitmaps.

Step 7. Sequentially take seven bits from the bitmaps and denote them as cover vector x. Then read three secret message bits and denote them as vector m each time.

Step 8. Perform the matrix embedding procedure which is the same as in Steps 2-3 in Phase 1 until the entire bit-stream in the bitmaps is successively processed.

Step 9. Update the bitmaps (BM'_1, BM'_2,..., BM'_k) and combine them with the corresponding quantization levels to obtain the final compressed trios (a^*, b^*, BM'_i).

After the whole embedding process is done, we successfully conceal the secret message into the compressed trios.

An embedding example is given in Figure 3 to demonstrate the embedding procedure of the proposed scheme. Denote that two compressed trios are (123, 156, 0111101101001011) and (86, 88, 0111101001010001), respectively. Assume that the secret message m is '0001101001011101100'. The parity check matrix H used in this example is shown in Equation (4), in which all the columns in H have been put in ascending order in decimal form.

$$\mathrm{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4}$$

Firstly, we embed three secret bits into the first quantization level pair (123, 156) according to the operations in Steps 1-4 of Phase 1. Then the difference value between a' and b' is calculated to check whether it is suitable for hiding one more bit. In this case, the pair belongs to the embeddable category. Since the secret bit is '1', their location will be interchanged. Then the second quantization level pair is operated in the similar way as the first pair. When embedding is finished according to Phase 1, we will hide secret message into bitmaps. The final results are illustrated in Figures 3(c) and (d), where the bold one indicates the alteration.

3.2 Extracting Phase

In the extracting phase, the secret message is extracted from a received watermarked AMBTC compressed code stream by using our designed rules. The flowchart of the extraction procedure is depicted in Figure 4. At the receiving side, the receiver firstly extracts the secret message hidden in Phase 1 from the two quantization level pairs by comparing their values. The absolute difference between a and b is computed to check whether it has been applied the interchange operation. If the absolute difference is larger than eight, we can ascertain that one more bit has been embedded into each quantization level pair and the secret bit is obtained by the inverse way as that of in the embedding phase. Otherwise, no secret bit is embedded by interchange operation. Then the rest secret data can be easily achieved by Equation (3) from the bitmaps. Finally, all bits are concatenated to exactly recover the hidden secret message.

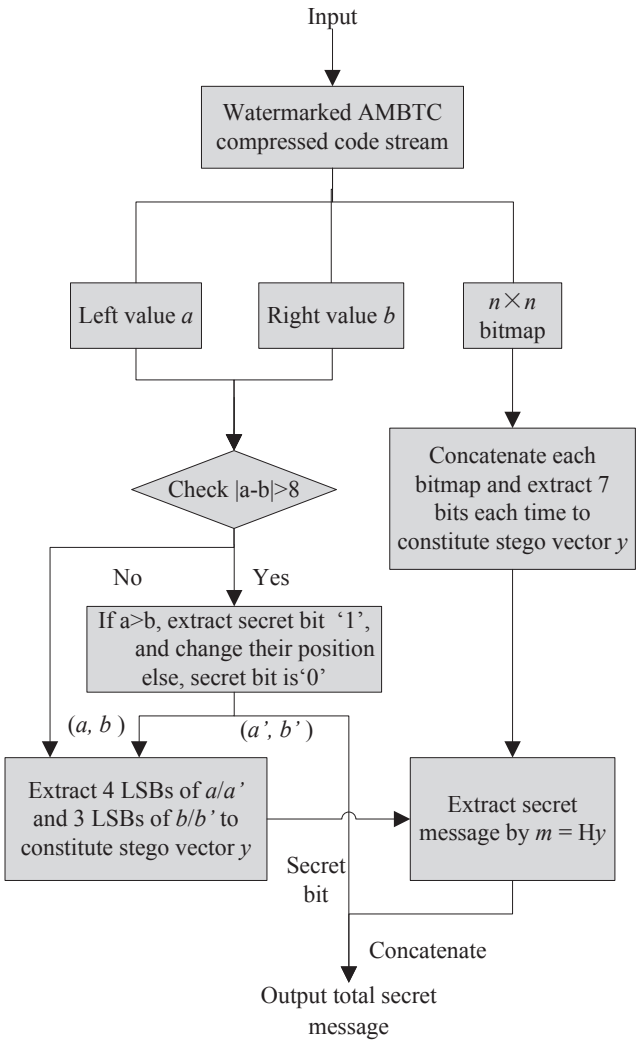


Figure 4: The flowchart of the extracting phase

4 Experimental Results

In this section, we conduct some experiments and discuss the results to demonstrate the superiority of our proposed scheme. Figure 5 lists the six grayscale test images used in our experiments, where they are 'Lena', 'Zelda', 'Elain', 'Jet', 'Boat', 'Goldhill', respectively. The block size used in the AMBTC compression is set as 4×4 in our experiment. We employ four parameters to measure the performance of the proposed scheme, where they are peak signal-to-noise ratio *PSNR*, hiding capacity *HC*, bit rate *BR*, and hiding efficiency *HE*. As we all know that

Table 1: The comparative results of the proposed scheme with other schemes

Scheme	Criteria	Lena	Zelda	Elain	Jet	Boat	Goldhill	Average
AMBTC	PSNR (dB)	33.23	36.74	33.83	33.25	31.76	32.87	33.61
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Proposed scheme	PSNR (dB)	28.92	32.25	29.41	28.11	27.56	28.74	29.17
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	169250	168480	172479	168388	170103	172771	170245
Chang et al's scheme [4]	PSNR (dB)	30.71	34.26	31.26	30.99	31.02	30.49	31.45
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	114688	114688	114688	114688	114688	114688	114688
Sun et al's scheme [13]	PSNR (dB)	Cannot be reconstructed by the watermarked/stego code stream						
	BR (bpp)	2.06	2.06	2.06	2.06	2.06	2.06	2.06
	HC(bits)	64008	64008	64008	64008	64008	64008	64008
Hong et al's scheme [7]	PSNR (dB)	33.23	36.74	33.83	33.25	31.76	32.87	33.61
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	16384	16384	16384	16384	16384	16384	16384
Chuang et al's scheme [3]	PSNR (dB)	30.45	33.98	31.07	30.52	29.84	30.52	31.06
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	13051	13169	13248	12637	11989	12486	12763

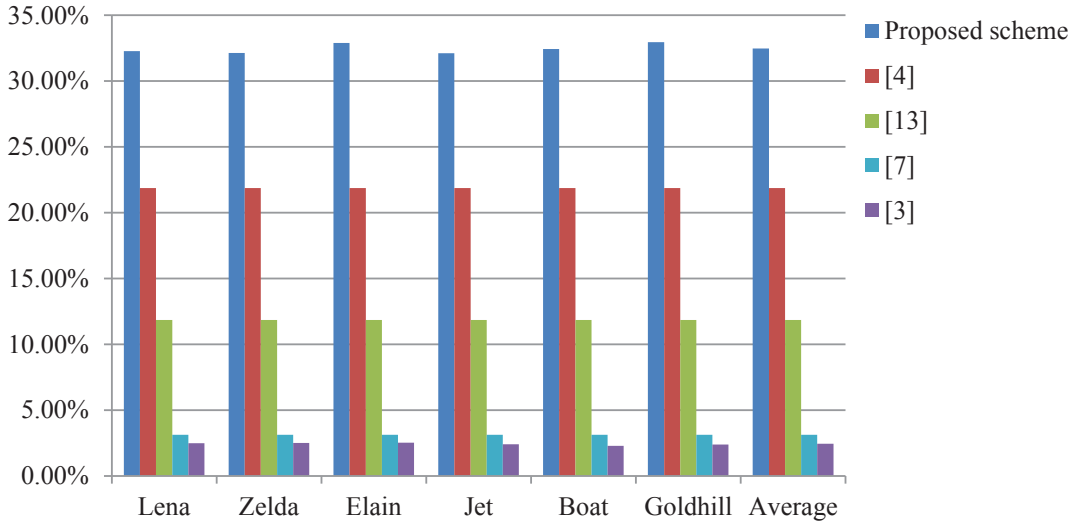


Figure 6: Comparisons of hiding efficiency between the proposed scheme and other schemes

peak signal-to-noise *PSNR* is used to evaluate the visual quality of the watermarked AMBTC compressed images. With respect to the value of *PSNR*, it is defined as shown in Equation (5), and MSE (mean square error) is calculated by Equation (6). Bit rate *BR* measures the number of bits required to represent one pixel, which is computed by Equation (7) and *CS* means the length of the output code stream. Hiding capacity *HC* is used to measure the number of embedded secret bits. Hiding efficiency *HE* is defined by Equation (8), which represents the percentage

(a) Lena (b) Zelda (c) Elain

(d) Jet (e) Boat (f) Goldhill

Figure 5: Test standard images

of *HC* and *CS*.

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})(\text{dB}), \quad (5)$$

$$MSE = \sum_{i=1}^{H}\sum_{j=1}^{W}(x_{i,j} - x'_{i,j})^2/(L \times W), \quad (6)$$

$$BR = \frac{|CS|}{L \times W}(bpp), \quad (7)$$

$$HE = \frac{HC}{|CS|}, \quad (8)$$

where $x_{i,j}$ and $x'_{i,j}$ are the values of pixels in the original grayscale image and the stego image, respectively.

We compare our scheme with the other four existing schemes [3, 4, 7, 13]. Table 1 shows the comparative results for different test images between the proposed scheme and other schemes under different criteria. With respect to the hiding capacity, we observe that the average hiding capacity of Chuang and Chang's scheme is the lowest among these schemes while the proposed scheme achieves the highest. The highest capacity reaches 172771 bits which is more than half of that of Chang et al.'s scheme. Although the payload is significantly increased with the cost of visual quality, the average *PSNR* of the proposed scheme is still greater than 29dB which is acceptable by human visual system (HVS). The visual quality of the proposed scheme decreases as the complexity of the texture increases, because the difference between low- and high-mean values is mostly bigger in complex images than that of in smooth images. The bit rate of the proposed scheme is the same as that of AMBTC algorithm. Compared with Sun et al.'s scheme, extra bits are introduced to be included in the watermarked code stream, which leads to a higher bit rate than the other four schemes. Furthermore, it cannot be constructed by the stego code stream since it contains extra data. Apparently, the proposed scheme outperforms other schemes in terms of hiding efficiency *HE* mainly due to its high embedding capacity. Figure 6 illustrates the comparative results of hiding capacity *HE* of the proposed scheme and the other schemes. It is notable that the *HE* of Chang et al.'s scheme is double that of Sun et al.'s scheme. However, the embedding efficiency of our scheme is 10% higher than that of Chang et al.'s scheme.

5 Conclusions

In this paper, a novel data hiding scheme based on (7, 4) Hamming code for AMBTC compressed images is presented. It achieves high embedding capacity while preserving an acceptable visual quality when compared with other existing BTC-based schemes. We not only hide secret message into two quantization levels but also embed secret bits into the bitmaps. For each embedding group consisting of three secret bits and seven cover bits, only one bite in the cover vector will be flipped and three secret bits are successfully embedded. Moreover, to further increase hiding capacity, we apply the interchange operation to the quantization pair to hide one extra bit. The experimental results confirm that the proposed scheme outperforms other existing schemes in terms of embedding capacity which can be used to the high-payload-needed application.

References

[1] P. K. Amin, N. Liu, K. P. Subbalakshmi, "Statistical attack resilient data hiding," *International Journal of Network Security*, vol. 5, no. 1, pp. 112–120, 2007.

[2] J. Chen, W. Hong, T. S. Chen, and C. W. Shiu, "Steganography for BTC compressed images using no distortion technique," *The Imaging Science Journal*, vol. 58, no. 4, pp. 177–185, 2010.

[3] J. C. Chuang and C. C. Chang, "Using a simple and fast image compression algorithm to hide secret information," *International Journal of Computers and Applications*, vol. 28, no. 4, pp. 329–333, 2006.

[4] C. C. Chang, Y. J. Liu and S. T. Nguyen, "A novel data hiding scheme for Block Truncation Coding - compressed images using dynamic programming strategy," in *Proceedings of 6th International Graphic and Image Processing*, pp. 94431Q–94431Q, 2015.

[5] R. Crandall, "Some notes on steganography," 1998. (`http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia/LINKS\%20LOCALI/matrix-encoding.pdf`)

[6] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, 1979.

[7] W. Hong, T. S. Chen and C. W. Shiu, "Lossless steganography for AMBTC compressed images," in *Proceedings of 1st International Congress on Image and Signal Processing*, vol. 2, pp. 13–17, Sanya, China, 2008.

[8] M. Jo, H. D. Kim, "A digital image watermarking scheme based on vector quantization," *IEICE Transactions on Information and System*, vol. E85-D, no. 6, pp. 1054–1056, 2002.
[9] C. Kim, et al. "A(2, 2) secret sharing scheme based on hamming code and AMBTC," in *Intelligent Information and Database Systems*, LNCS 7198, pp. 129–139, 2012.
[10] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
[11] M. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1148–1157, 1984.
[12] Q. Mao, "A fast algorithm for matrix embedding steganography," *Digital Image Processing*, vol. 25, no. 4, pp. 248–254, 2014.
[13] W. Sun, et al. "High performance reversible data hiding for block truncation coding compressed image," *Signal, Image and Video Processing*, vol. 7, no. 2, pp. 297–306, 2013.
[14] N. I. Wu, C. M. Wang, M. S. Hwang, "Data Hiding: Current Status and Key Issues," *International Journal of Network Security*, vol. 4, no. 1, pp. 1–9, 2007.
[15] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
[16] F. J. M. Williams and N. J. Sloane, "The Theory of Error-Correcting Codes," *Elsevier*, 1977.
[17] C. Zhu, L.H. Li, T.J. Wang, Z.Y. He, "Partial-distortion-weighted fuzzy competitive learning algorithm for vector quantization," *Electronics Letters*, vol. 30, no. 6, pp. 505–506, Mar. 1994.
[18] C. Zhu and L. M. Po, "Partial distortion sensitive competitive learning algorithm for optimal codebook design," *Electronics Letters*, vol. 32, no. 19, pp. 1757–1758, 1996.
[19] C. Zhu, L.M. Po, "Minimax partial distortion competitive learning for optimal codebook design", *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1400–1409, 1998.
[20] C. Zhu and Y. Hua, "Image vector quantization with minimax L_∞ distortion," *IEEE Signal Processing Letters*, vol. 6, no. 2, pp. 25–27, 1999.
[21] S. Zhang, T. Gao, L. Yang, "A reversible data hiding scheme based on histogram modification in integer DWT domain for BTC compressed images," *International Journal of Network Security*, vol. 18, no. 4, pp. 718–727, 2016.

Junlan Bai received the B.S. degree in electronic engineering from University of Electronic Science and Technology of China, China, in 2013. She is currently pursuing the M.S. degree at the same university. Her research interests include information hiding and digital image processing.

Chin-Chen Chang received his Ph.D. degree in computer engineering 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 Visiting 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 organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. 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.

Data Encryption Scheme Based on Rules of Cellular Automata and Chaotic Map Function for Information Security

Warakorn Srichavengsup and Wimol San-Um
(Corresponding author: Warakorn Srichavengsup)

Intelligent Electronic System Research Laboratory, Faculty of Engineering, Thai-Nichi Institute of Technology
1771/1 Pattanakarn Road, Suanluang, Bangkok 10250
(Email: warakorn@tni.ac.th)

(Received Sept. 11, 2015; revised and accepted Dec. 7, 2015 & Jan. 15, 2016)

Abstract

Cryptography has recently played a significant role in secure data transmissions and storages. Most conventional data encryption schemes are relatively complicated and complexity in encrypted keys is insufficient, resulting in long computational time and low degree of security against all types of attacks. Consequently, a highly-secured and robust data encryption scheme is necessary. This paper therefore presents the data encryption scheme based on a combination of Cellular Automata (CA) and a robust chaotic system that employs absolute-value piecewise-linear nonlinearity. The proposed encryption scheme is not only applicable to image encryption but also extendable to text and Excel files. Simulation results reveal that the entropy of the encrypted image is nearly 8 corresponding to the image entropy theory, and the correlation coefficients of before-and-after encrypted images are close to zero. Besides, the histogram of the encrypted image of pixel values in the range (0-255) is apparently flat, indicating that the color in image is fully distributed. Such results indicate that the proposed data encryption scheme offers a high level of security with fast processing time duration. The implementation of image encryption Android application is demonstrated. Comparisons to other related approaches are also included.

Keywords: Cellular automata, chaotic map, data encryption, data security

1 Introduction

Advances in communications have led to great demand for secured data transmissions [7, 11] and storage for a variety of applications such as medical, industrial and military systems. The secured data transmissions greatly require reliable, fast and robust security systems, and can be achieved through cryptography, which is a technique of information privacy protection under hostile conditions [17]. Data and image cryptography may be classified into two categories, i.e. (1) pixel value substitution which focuses on the change in pixel values so that original pixel information cannot be read, and (2) pixel location scrambling which focuses on the change in pixel position. Conventional cryptography such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES), and RSA algorithm may not be applicable in real-time image encryption due to high computational time and high computing power, especially for the images with large data capacity and high correlation among pixels [10].

The utilization of chaotic systems has broadly been suggested as one of potential alternative encryption techniques in secured data and image transmissions [1, 8]. In comparison to conventional encryption algorithms, chaos-based encryptions are sensitive to initial conditions and parameters while conventional algorithms are sensitive to designated keys. Furthermore, chaos-based encryptions spread the initial region over the entire phase space, but cryptographic algorithms shuffle and diffuse data by rounds of encryption [2]. Therefore, the security of chaos-based encryptions is defined on real numbers through mathematical models of nonlinear dynamics while conventional encryption operations are defined on finite sets. Such chaos-based encryption aspects consequently offer high flexibility in encryption design processes and acceptable privacy due to vast numbers of chaotic system variants and numerous possible encryption keys.

Chaos-based encryption algorithms are performed in two stages, i.e. the confusion stage that permutes the image pixels and the diffusion stage that spreads out pixels over the entire space. Most existing chaos-based encryptions based on such two-stage operations employ both initial conditions and control parameters of 1D, 2D, and 3D chaotic maps such as Baker map [23, 27], Arnold cat map [14, 24], and Standard map [12, 25] for secret key

generations. Furthermore, the combinations of two or three different maps have been suggested [4, 6] in order to achieve higher security levels. Despite the fact that such maps offer satisfactory security levels, iterations of maps require specific conditions of chaotic behaviors through a narrow region of parameters and initial conditions. Consequently, the use of iteration maps has become typical for most of proposed ciphers and complicated techniques in pixel confusion and diffusion are ultimately required.

Cellular Automata is a discrete system, which has been proven to be useful in the models of complexity and nonlinear dynamic systems. It consists of a set of cells and a new state of each cell depends on the rule number and the state of neighboring cells. Stephen Wolfram [26] initially employed a CA with the rule 30 to produce the pseudo-random number sequences, and extensive CA rules have been employed lately for data encryption [5, 15, 22].

As for compact and effective chaos-based data and image encryption, this paper presents a high-level security, very large key space and robust digital image encryption through the use of cellular automata sequences combined with chaotic systems. The proposed chaotic map uses absolute-value piecewise-linear nonlinearity and offers robust chaotic regions over broad parameter spaces with high degree of randomness through chaoticity measurements using the Lyapunov exponent. Experiments have been performed in MATLAB using standard color images. Nonlinear dynamics of the chaotic maps are initially investigated in terms of Cobweb map, chaotic attractor, Lyapunov exponent spectrum, bifurcation diagram, and 3-dimensional parameter spaces. Qualitative mesures of encryption performances are evaluated through pixel density histograms, 3-dimensional power spectral density, key space analysis, key sensitivity, and correlation plots. Additionally, quantitave measures of encryption performances are also indicated by correlation coefficients, NPCR and UACI. Practical application in Adndroid devices with correct-key and wrong-key decryptions are also demonstrated.

2 Detailed Descriptions of Proposed Chaotic Map and Cellular Automata

2.1 Proposed Chaotic Map

Chaotic system is typically a dynamic system that possesses some significant properties, involving the sensitive dependence on initial conditions and system parameters, the density of the set of all periodic points, and topological transitivity. Of a particular interest, a chaotic map is the lowest one-dimensional evolution function in discrete-time domain that exhibits chaotic behaviors. In general, chaotic systems reveal two types of chaotic attractors, i.e. (i) a fragile chaos in which the attractors disappear with perturbations of a parameter or coexist with other attractors, and (ii) a robust chaos, which is defined by the absence of periodic windows and coexisting attractors in some neighborhood of the parameter space. This paper alternatively proposes a mathematically simple chaotic map with robust chaos through the use of absolute-value piecewise-linear nonlinearity expressed as

$$x_{n+1} = |px_n - q| \tag{1}$$

As will be seen later, such mathematical simplicity of the proposed chaotic map in Equation (1) offers robustness that has no sensitivity on the change of system parameters. Investigations on chaotic behaviors of chaotic maps of Equation (1) can be achieved qualitatively and quantitatively through a bifurcation diagram and the Lyapunov Exponent (LE), respectively. The bifurcation diagram indicates possible long-term values, involving fixed points or periodic orbits, of a system as a function of a bifurcation parameter. The stable solution is represented by a straight line while the unstable solutions are generally represented by dotted lines, showing thick regions. On the other hand, the LE is defined as a quantity that characterizes the rate of separation of infinitesimally close trajectories and is expressed as

$$LE = \lim_{n\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log_2 \frac{dx_{n+1}}{dx_n} \tag{2}$$

where N is the number of iterations. Typically, the positive LE indicates chaotic behaviors of dynamical systems and the larger value of LE results in higher degree of chaoticity. Dynamic properties can be described in terms of Cobweb plots, bifurcations, Lyapunov exponents, and chaotic waveforms in time domain. The system in Equation (1) possesses two fixed points P_{x1} and P_{x2} as follows

$$P_{x1} = \frac{q}{p-q} \text{ and } P_{x2} = \frac{q}{p+q} \tag{3}$$

The corresponding Jacobian is given by

$$|p \text{ sign}\,(q + px)| \tag{4}$$

The map has the only two parameters p and q that set dynamic properties of the systems. Simulation results have been performed using MATLAB with the initial condition of x_0=0.01. Figure 1 shows the Cobweb plots of the proposed chaotic map where the iterations are dense corresponding to the two fixed points described in Equation (3). Figures 2 and 3 show the bifurcation diagram and the LE spectrum of Equation (1) where p is in the region [0-2] and q is set to be 2. It is apparent in Figures 2 and 3 that there are no periodic windows appear, i.e. smooth chaos, in the bifurcation diagram and the LE spectrum is smoothly varied corresponding to the bifurcation diagram. Figure 4 shows an apparently chaotic waveform in time-domain.

2.2 Cellular Automata

Cellular Automata (CA) were first devised by Stanislaw Ulam and John von Neumann in the 1940s. Stephen Wolfram published a book entitled "A New Kind of Science"

Table 1: All possible neighborhoods and the outcome of rule 30

left	center	right	outcome
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

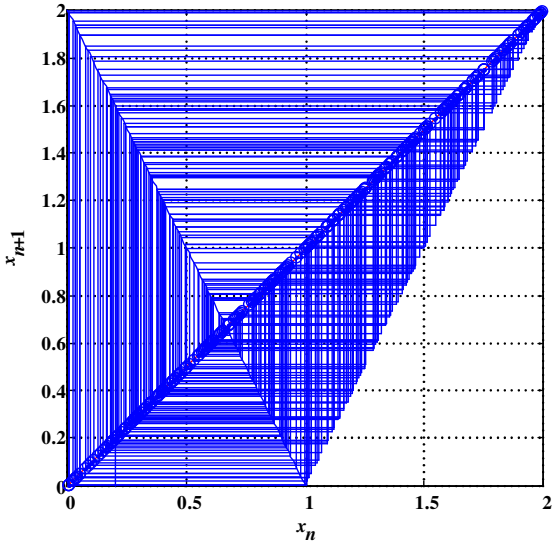


Figure 1: The Cobweb plots of the proposed chaotic map

in 2002, mentioning that cellular automata can be applied in many areas of science, including computer processors and cryptography. Elementary Cellular Automata are composed of cells positioned in a grid, each cell change a state depending on the states of its neighboring cells. For example, the outcomes of all possible neighborhoods for the rule 30 ($30 = 00011110_2$) are illustrated in Table 1 and Figure 5.

Starting with a single black cell, the first 12 steps of the evolution for rule 30 are demonstrated in Figure 6. The produced patterns of CA with some specific rules are shown in Figure 7. It can be noticed that some rules such as rules 30 and 101 potentially produce the chaotic behaviors which can be employed in the cryptography. The results of [20] concluded that rules 30, 86, 90, 101, 105, 150, 153, 165 are able to generate pseudorandom number sequences of a very high randomness quality and the CA-based system is very robust against the attempts of breaking the cryptography key.

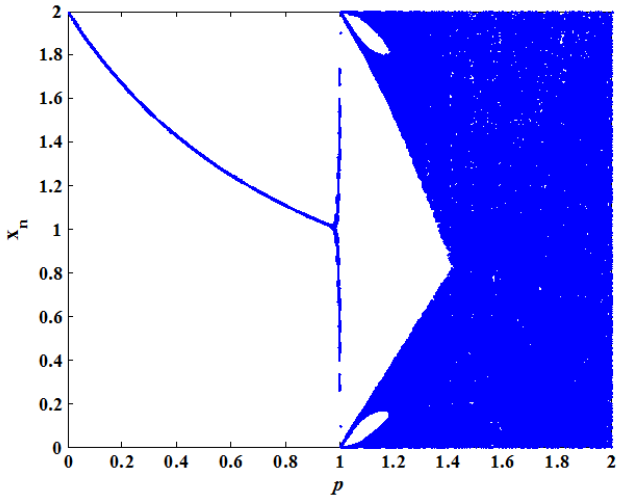


Figure 2: The bifurcation diagram in the range p=[0,2]

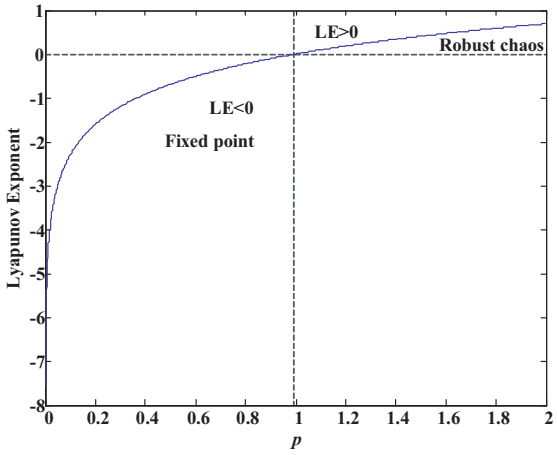


Figure 3: The LE spectrum in the range p=[0,2]

3 Proposed Data Encryption Using Chaotic Map and Cellular Automata

The whole structure of the proposed digital image encryption scheme using chaotic map and cellular automata is shown in Figure 8. The procedures for this encryption are described as follows:

1) The keys of the proposed scheme include parameters, involving initial conditions and system constants.

2) The original image is decomposed into red, green and blue components and each component is converted into binary format.

3) $W \times H$ iterations are required for the absolute chaotic map in order to generate the chaotic matrix X, where W and H are the width and height of original image in binary format. The $n{+}1^{th}$ element of the chaotic matrix can be calculated as in Equation (1) where

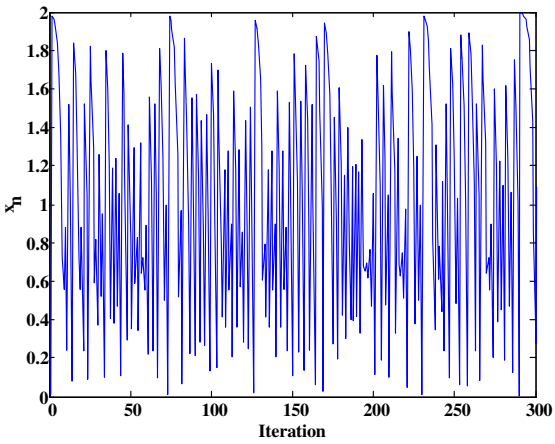


Figure 4: The apparently chaotic waveforms in time-domain

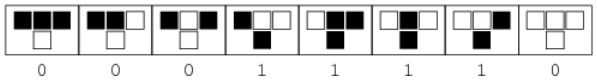


Figure 5: The outcome for every possible neighborhoods for rule 30

n=1, 2, ..., $W \times H$ and the value of p is given by

$$p = p_0 + \frac{\sum_{i,j,k} O(i,j,k)}{W \times H \times 255 \times 3 \times 10^6} \quad (5)$$

where the value p_0 is in a range of 1.999998 to 1.999999. Let $O(i,j,1)$, $O(i,j,2)$ and $O(i,j,3)$ be the intensity level of $(i,j)^{th}$ pixel of red, green and blue components, respectively.

4) X is transformed into a chaotic matrix with 2-dimension of $W \times H$.

5) The red, green and blue components and the chaotic matrix are combined through bitwise XOR operations.

6) The CA rule number and the bit sequence of the first row of CA matrix are chosen. Then the CA matrix is generated.

7) The encrypted red, green, and blue components are obtained by operating bitwise XOR on the two dimensional cellular automata matrix and the outcome of Step 5.

8) Finally, the encrypted red, green, and blue components are combined to form the encrypted image.

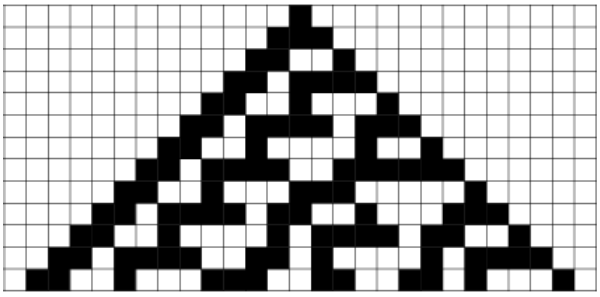


Figure 6: The first 12 steps of the evolution for rule 30

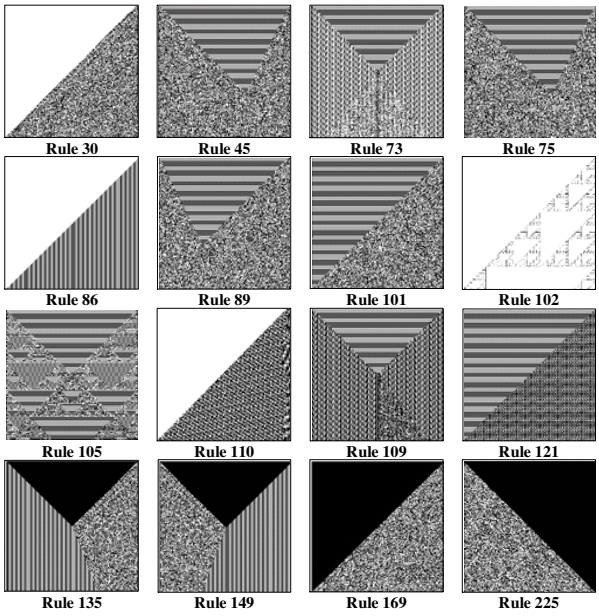


Figure 7: The produced patterns of CA rules

The decryption procedure is similar to that of encryption demonstrated above with reverse of encrypted image as input instead of original image in the encryption procedure.

The proposed encryption scheme can also be applied to any other data type such as plaintext as shown in Figure 9. The plaintexts in excel file and text file are illustrated in Figures 10 and 12. After encryption, the encrypted data in excel and text files can be shown in Figures 11 and 13.

4 Security Analysis

In order to evaluate the security of the proposed scheme, the key space analysis, histogram analysis, the correlation coefficient analysis of two adjacent pixels and differential attack analysis are performed.

4.1 Sensitivity Analysis

An ideal image encryption procedure should be sensitive with respect to the secret key, i.e., the change of a single bit in the secret key should produce a completely different encrypted image. The following experiments and results show key sensitivity of the presented scheme. An original image illustrated in Figure 14 is encrypted by using the correct key and the encrypted image is shown in Figure 15. Figure 16 illustrates the decrypted image using the right key. If there is only one bit difference between the encryption and decryption keys, an unexpected image

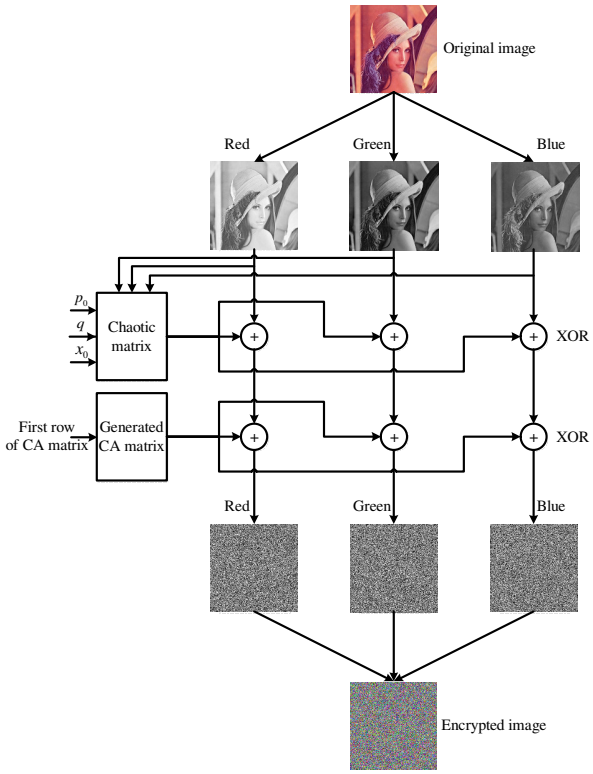


Figure 8: Proposed digital image encryption

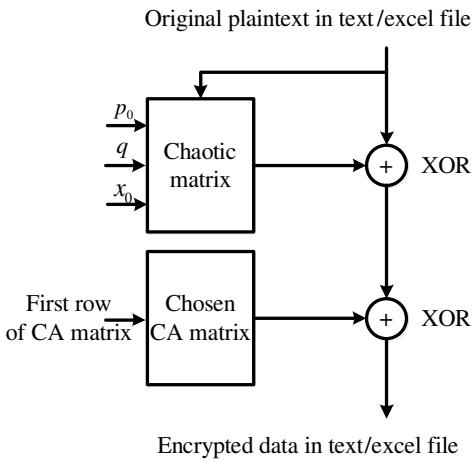


Figure 9: Proposed plaintext encryption in text/excel file

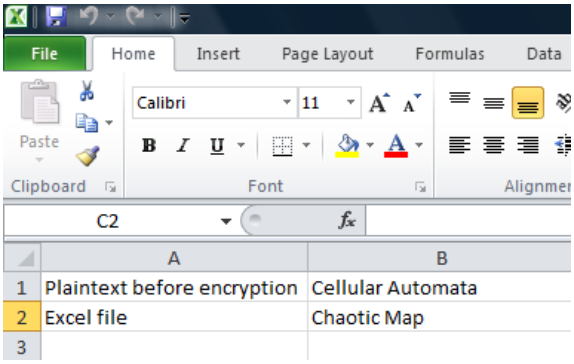


Figure 10: Plaintext before encryption in excel file

will be obtained as illustrated in Figure 17. So it can be concluded that the proposed encryption scheme is highly sensitive to the keys.

4.2 Key Space Analysis

Key space for the scheme means the number of all possible keys that can be adopted to encrypt data. Key space size ought to be sufficiently large, making brute-force attacks infeasible. For the proposed scheme, the initial conditions such as values of p_0, q and x_0, CA rule number and the bit sequence of the first row of CA matrix are used as keys. If the precision is 10^{-12} and as described before there are 8 possible CA rules that can be used for encryption such that the key space contains $10^{12}\times10^{12}\times10^{12}\times8\times2^{w}$ or $2^{w+3}\times10^{36}$ possible keys, where w, the length of the bit sequence of the first row of CA matrix, is equal to the width of original image in binary format W. As can be seen, the key space size is large enough to withstand the brute-force attacks.

4.3 Histogram Analysis (Histograms of Encrypted Image)

Histogram is a useful tool that displays the tonal distribution of a digital image. It illustrates the number of pixels at each intensity level. The histograms of red, green, and blue components of original and encrypted images are demonstrated in Figure 18. It can be noticed that particular intensity levels are dominant in the original images whereas the intensity levels of the encrypted image are uniformly distributed on [0, 255]. Consequently, it does not provide any information to perform any statistical analysis attack on the encrypted image.

4.4 3D Power Spectral Analysis

Discrete Fourier Transform (DFT) analysis can be used to attained the 3D power spectrum and the power spectral density is given by [18].

$$P(u,v)=\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} I(x,y)\cdot\exp(-j(2\pi/W)ux)\cdot\exp(-j(2\pi/H)vy) \quad (6)$$

where (x,y) represents the coordinate of pixels in the image, W and H are width and height of the image, $I(x,y)$ is intensity value of image at (x,y). The power spectral densities of the original and encrypted images are demonstrated in Figure 19. The original image has a peak power spectral density at the center while the power spectral density of the encrypted image is flat. The results indicate that the intensity values of the encrypted image are uniformly distributed all over the intensity range. This means that the encryption scheme is secure, as there is no information leakage.

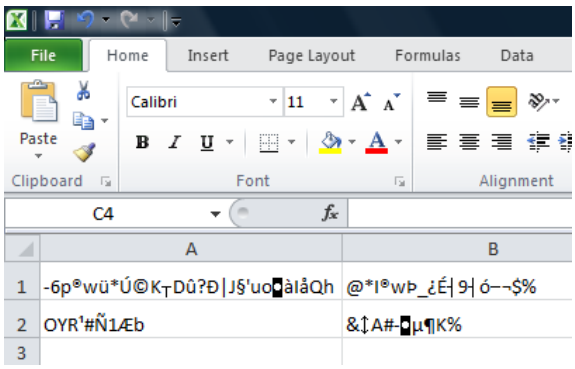


Figure 11: Encrypted data in excel file

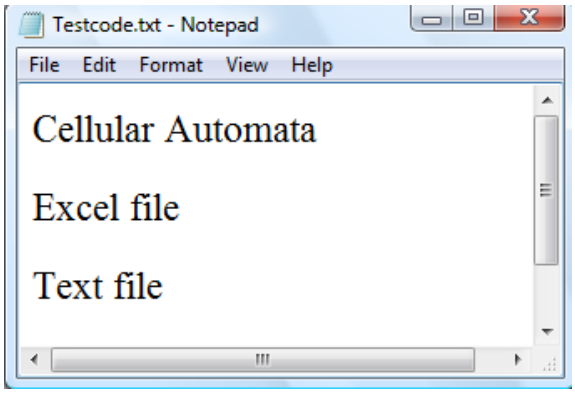


Figure 12: Plaintext before encryption in text file

4.5 Correlation Coefficient Analysis of Two Adjacent Pixels

A correlation is a statistical method that is used to measure degree of similarity between pairs of variables. In order to illustrate the relationship between two adjacent pixels in the digital image, correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two diagonally adjacent pixels in the original and encrypted images are analyzed.

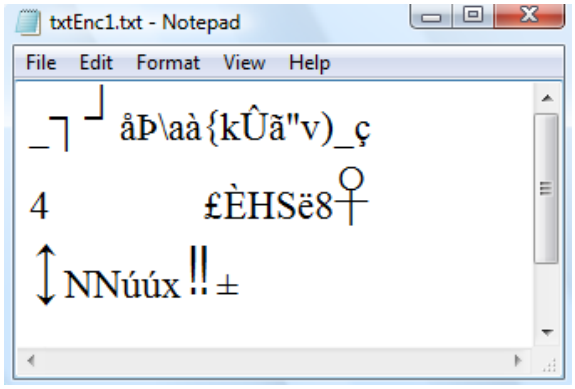


Figure 13: Encrypted data in text file

Figure 14: Original image

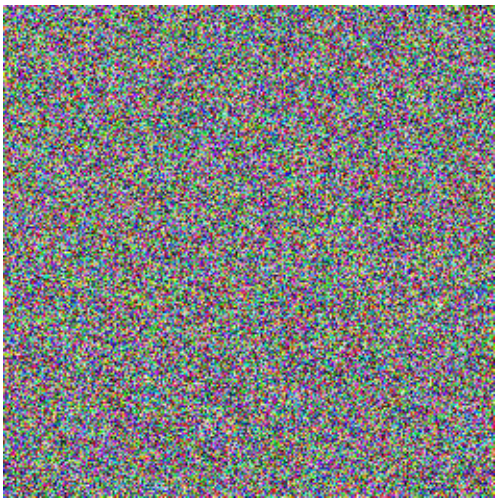


Figure 15: Encrypted image

The correlation coefficient is computed as follows:

$$C_r = \frac{N\sum_{i=1}^{N}(x_i \times y_i) - \sum_{i=1}^{N} x_i \times \sum_{i=1}^{N} y_i}{\sqrt{\left(N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2\right)\left(N\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2\right)}} \quad (7)$$

where x_i and y_i are the intensity level of two adjacent pixels and N is overall number of pixels in the digital image. Figure 20 illustrates the correlation distribution of two horizontally adjacent pixels in the original image and encrypted image. It can be noticed that the adjacent pixels of all encrypted images are highly unrelated demonstrated by scatter graphs. The correlation coefficients are illustrated in Table 2. As can be seen, the value of correlation coefficient of the encrypted image is nearly zero. This reveals that two adjacent pixels are extremely unrelated.

Table 2: Simulated correlation coefficient of the original and encrypted images

Correlation of Images	Correlation Coefficient Values
C_{RR}	-0.0020
C_{RG}	0.0050
C_{RB}	-0.0006
C_{GR}	-0.0015
C_{GG}	0.0009
C_{GB}	0.0009
C_{BR}	-0.0020
C_{BG}	0.0002
C_{BB}	0.0006

Table 3: Summary of NPCR and UACI tests

Measures	Proposed scheme	2D Baker map	DES
NPCR (red)	99.6155	99.5132	0.0045
NPCR (green)	99.6124	99.5407	0.0045
NPCR (blue)	99.6094	99.5849	0.0030
UACI (red)	33.3399	32.1693	0.0012
UACI (green)	33.3458	32.1788	0.0026
UACI (blue)	33.2698	32.3173	0.0089

Figure 16: Decrypted image using the right key

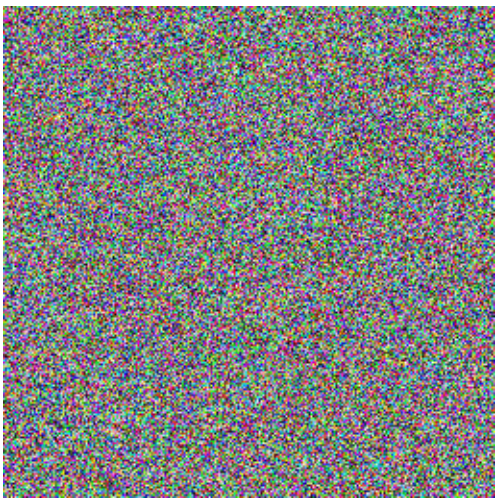


Figure 17: Decrypted image using the wrong key

4.6 Analysis of Differential Attack

In order to observe a relationship between the original image and the encrypted image, the rival may alter only one pixel of the original image, and then notices the difference of the outcome. A substantial change of the outcome is expected to make this differential attack infeasible. To see the impact of altering one pixel in plaintext image on the encrypted image. Two most common standards, Number of Pixel Change Rate (NPCR) and Unified Average Change Intensity (UACI), are used to evaluate the resistance against the differential attack. The NPCR determines the percentage of changed pixels between two encrypted images. The UACI measures the mean intensity of distinctions between two encrypted images. These two standards can be calculated as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (8)$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j}\left|\frac{C_1(i,j) - C_2(i,j)}{255}\right|\right] \times 100\% \quad (9)$$

Let $C_1(i,j)$ and $C_2(i,j)$ be $(i,j)^{th}$ pixel of two differ-

Table 4: Information entropy test

Test item	Proposed scheme	2D Baker map	DES
Information entropy	7.999223	7.999158	7.999096

Table 5: NIST statistical test suite results for one hundred 1 M-bit sequences generated using randomly initial condition

Statistical test	p-value	Pass rate
Frequency	0.537894	1.00
Block frequency	0.919617	0.99
Runs	0.747165	0.98
Long runs of ones	0.061725	0.99
Rank	0.490471	0.97
Discrete Fourier Transform (Spectral)	0.912314	1.00
Non overlapping templates	0.113991	0.96
Overlapping templates	0.022266	0.99
Universal	0.666870	0.98
Linear complexity	0.232466	0.98
Serial 1	0.741995	1.00
Serial 2	0.677427	1.00
Approximate entropy	0.444053	0.96
Cumulative sums (forward)	0.642075	0.96
Cumulative sums (reward)	0.493507	0.96
Random excursions	0.465971	0.97
Random excursions variant	0.307470	0.97

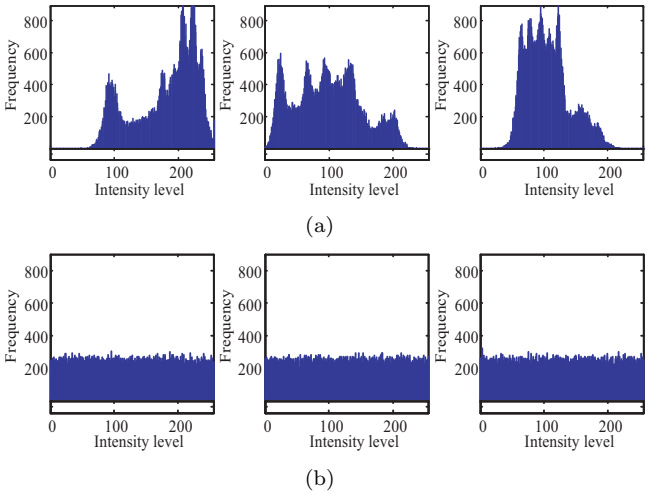


Figure 18: Histograms of RGB components: (a) Original image, (b) Encrypted image

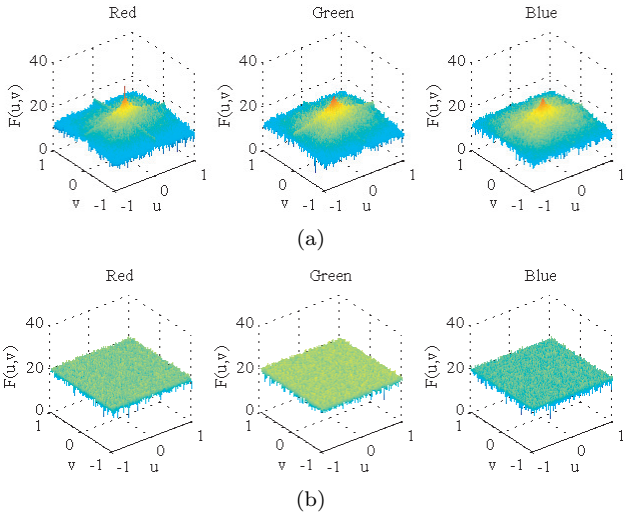


Figure 19: 3D power spectral density: (a) Original image, (b) Encrypted image

ent encrypted-images. The parameters W and H are the width and height of encrypted-images and $D(i,j)$ is defined as follows:

$$D(i,j) = \begin{cases} 1, & if C_1(i,j) \neq C_2(i,j) \\ 0, & if C_1(i,j) = C_2(i,j) \end{cases} \quad (10)$$

The experimental results comparison with 2D Baker map and DES schemes are illustrated in Table 3. The obtained values obviously demonstrate that changing of one pixel in the original image leads to a substantial change in the encrypted image, therefore the proposed scheme is secure against differential attacks.

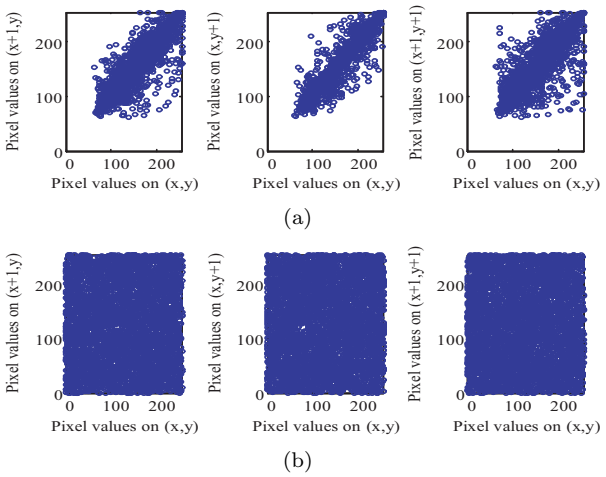


Figure 20: Image correlation experiments comprising horizontally, vertically, and diagonally adjacent pixels: (a) Original image, (b) Encrypted image

4.7 Information Entropy Analysis

Information entropy can be determined as a probabilistic measure of uncertainty associated with a random variable. It can be calculated as follows:

$$H(S) = \sum_{S} P(s_i) \log_2 \frac{1}{P(s_i)} \tag{11}$$

$P(s_i)$ denotes the probability of symbol s_i. The test results comparison with 2D Baker map and DES schemes are demonstrated in Table 4. The entropy value acquired approaches the theoretical value of 8, demonstrating that the proposed scheme is secure against the attacks.

4.8 NIST Statistical Tests

NIST statistical tests [16] are employed for examining the random sequences. The NIST test suite is a statistical package composing of 16 tests that are used to test the randomness of binary sequences generated by software or hardware. These tests concentrate on a various types of non-randomness that may appear anywhere in a sequence. For each of these tests, one hundred sequences of length 10^6 bits are tested. In accordance with NIST document, a pass rate of 96% is satisfactory. A test results are shown in Table 5. The achieved results illustrate the supreme statistical properties of the generated random sequences.

4.9 Flexibility and Speed Analysis

The proposed encryption scheme is flexible since there is no restriction on the size of the original data and it can encrypt many different types of data, such as plaintext, binary data and digital image. The running speed of the encryption scheme, on a 3.20 GHz Intel(R) Xenon(R) computer with 16 GB of RAM running Windows 7 (64-bit) is about 9.3 Mb/s (Megabits per second).

5 Comparison of Proposed Scheme with Other Existing Techniques

The proposed scheme is analyzed and compared with the existing (a) chaotic map and (b) chaotic flow in terms of numbers of terms in equation, calculation time, text encryption and decryption, type of characters, key space, the positive LE value (LE $>$ 0), key sensitivity, robust chaos and power spectrum, as illustrated in Table 6. From Table 6, it can be concluded that the proposed scheme offers better aspects and performances than other chaotic schemes. For instance, the computational time of the proposed scheme is relatively fast due to low-dimension of system order comparing to the 3rd-order chaotic flows that require particular algorithms to solve for ordinary differential equations. The proposed scheme can be applied to text encryption and decryption based on Unicode system and ASCII. Key space is equal to $2^{w+3}\times10^{36}$ digits, where w is the length of the bit sequence of the first row of CA matrix. The key space is large enough to make the attack infeasible. The system is truly chaotic (LE $>$ 0) with robust chaos and sensitive to keys. Moreover, the power spectral density of the encrypted data is uniformly distributed.

6 Security Implementation

As for illustration, the proposed image encryption scheme will be implemented on an Android device in order to protect the important and confidential data. Figure 21 demonstrates Android application user interface for the encryption. The procedures for the image encryption are described as follows:

1) Input the system parameters (p_0, q and x_0).
2) Locate the image file (.PNG) of the initial bit sequence of the first row of CA matrix. This bit sequence is saved in PNG format for the convenience of both sender and receiver.
3) Choose and import the image to be encrypted.
4) Click on "ENCRYPT" button.
5) The results of the encryption are displayed on the screen.

The initial conditions and the image file (.PNG) of the initial bit sequence of the first row of CA matrix are shared between sender and receiver. The sender uses the shared key for encryption and the receiver uses the same shared key for decryption.

Android application user interface for the decryption is illustrated in Figure 22. The procedures for the image decryption are described as follows:

1) Input the system parameters (p_0, q and x_0).

Table 6: Performance comparison between the proposed scheme and other chaotic schemes, "/" is "Yes", "x" is "No" and "-" is "not presented in the paper".

Comparisons	Chaotic types					
	Chaotic maps		Chaotic flows			
	Proposed scheme	Logistic [3]	Jerk [21]	Lu [13]	Lorenz [9]	Rossler [19]
Terms in equation	2	2	5	6	7	7
Computational time	Fast	Fast	Low	Low	Low	Low
Text encryption/decryption	/	/	x	x	/	x
Type of characters	Unicode and ASCII	ASCII	x	x	ASCII	x
Key space	$2^{w+3}\times10^{36}$ digits	-	-	-	-	-
LE > 0	/	/	/	/	/	/
Robust chaos	/	x	x	x	x	x
Power spectrum	/	x	x	x	x	x
Key sensitivity	High	x	x	x	x	x

2) Locate the image file (.PNG) of the initial bit sequence of the first row of CA matrix.

3) Choose and import the encrypted image to be decrypted.

4) Click on "DECRYPT" button.

5) The results of the decryption are displayed on the screen.

These results confirm that the proposed encryption scheme can protect the important and confidential data on Android device.

7 Conclusions

The data encryption scheme based on rules of Cellular Automata and chaotic map function has been proposed. The proposed chaotic map exploits absolute-value piecewise-linear nonlinearity that offers robust chaotic regions over broad parameter spaces with high degree of randomness. Cellular Automata has also been used to generate pseudo-random number sequences with a very high randomness. A combination of cellular automata sequences and chaotic system has been realized in order to achieve a high level of security and adequately large key space. The proposed scheme is flexible since it can take the original data of any length and encrypt many types of data, such as plaintext, binary data and digital image. Experimental results reveal that the proposed scheme has many important features including: (i) high sensitive to the key and original message, (ii) large key space, (iii) resistant to various attacks such as the brute-force, statistical and differential attacks, and (iv) high data encryption speed. These properties make the proposed data encryption scheme to be suitable for real-time implementation as demonstrated in smart phone with Android operating system.

Acknowledgments

The authors are grateful to Thai-Nichi Institute of Technology for financial supports. The authors would also like to thank Mr. Sivapong Nilwong for his useful suggestions.

References

[1] C. C. Chang, Y. Liu, G. Song, Y. Liu and D. Wang, "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.

[2] G. Chen, Y. Mao and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals*, vol. 21, pp. 749–761, 2004.

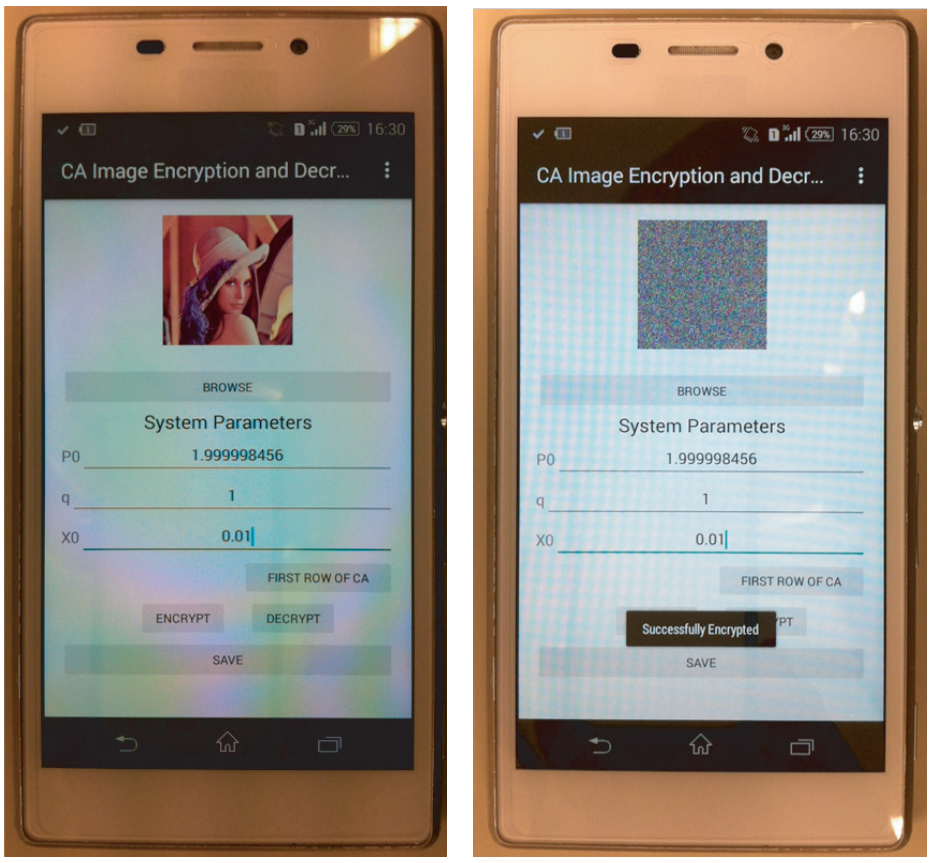
[3] C. K. Chen, C. L. Lin and Y. M. Chiu, "Text encryption using ECG signals with chaotic Logistic map," *IEEE International Conference on Industrial Technology*, pp. 1741–1746, 2010.

[4] K. Gupta and S. Silakari, "New approach for fast color image encryption using chaotic map," *Journal of Information Security*, vol. 2, no. 4, pp. 139–150, 2011

[5] S. U. Guan, S. Zhang, and M. Quieta, "2-D CA variation with asymmetric neighborship for pseudo-random number generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 3, pp. 378–388, 2004.

[6] F. Huang and Y. Feng, "Security analysis of image encryption based on two-dimensional chaotic maps and improved algorithm," *Frontiers of Electrical and Electronic Engineering in China*, vol. 4, no. 1, pp. 5–9, 2009.

[7] H. F. Huang, P. H. Lin and M. H. Tsai, "Convertible multi-authenticated encryption scheme for data communication," *International Journal of Network Security*, vol. 17, no. 1, pp. 40–48, 2015.



(a) (b)

Figure 21: Android application user interface: (a) Original image for the encryption process, (b) The encrypted image

[8] I. A. Ismail, M. Amin and H. Diab, "A digital image encryption algorithm based a composition of two chaotic Logistic maps," *International Journal of Network Security*, vol. 11, no. 1, pp. 1–10, 2010.

[9] Y. Ji, C. Wen and Z. G. Li, "A practical chaotic secure communication scheme based on Lorenz model," *Proceedings of the 4th International IEEE Conference on Industrial Informatics (INDIN'06)*, pp. 576–580, 2006.

[10] G. H. Karimian, B. Rashidi and A. Farmani, "A high speed and low power image encryption with 128-bit AES algorithm," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 3, pp. 367, 2012.

[11] C. C. Lee, S. T. Chiu and C. T. Li, "Improving security of a communication-efficient three-party password authentication key exchange protocol," *International Journal of Network Security*, vol. 17, no. 1, pp. 1–6, 2015.

[12] S. Lian, J. Sun and Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons and Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[13] J. Lu, G. Chen and S. Zhang, "The compound structure of a new chaotic attractor," *Chaos, Solitons and Fractals*, vol. 14, no. 5, pp. 669–672, 2002.

[14] X. Ma, C. Fu, W. Lei and S. Li, "A novel chaos-based image encryption scheme with an improved permutation process," *International Journal of Advancements in Computing Technology*, vol. 3, no. 5, pp. 223–233, 2011.

[15] S. Nandi, B. K. Kar and P. P. Chaudhuri., "Theory and applications of cellular automata in cryptography," *IEEE Transactions on Computers*, vol. 43, no. 12, pp. 1346–1357, 1994.

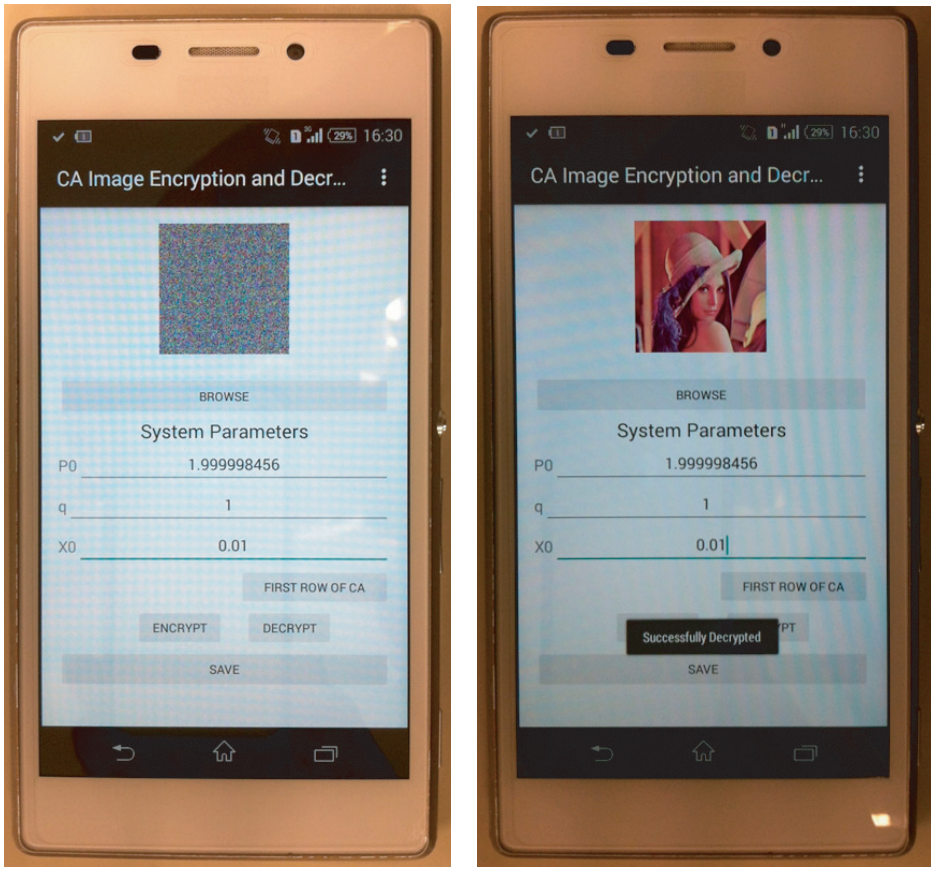
[16] NIST Special Publication 800-22 rev1, A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications, online document, 2008. (<http://www.csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html>)

[17] M. Philip, "An enhanced chaotic image encryption," *International Journal of Computer Science*, vol. 1, no. 5, pp. 201, 2011.

[18] Z. Peng, T. B. Kirk, "Two-dimensional fast Fourier transform and power spectrum for wear particle analysis," *Tribology International*, vol. 30, no. 8, pp. 583–590, 1997.

[19] O. E. Rössler, "An equation for continuous chaos," *Physics Letters A*, vol. 57, no. 5, pp. 397–398, 1976.

[20] F. Seredynski, P. Bouvry and A. Y. Zomaya, "Cellular programming and symmetric key cryptography



(a) (b)

Figure 22: Android application user interface: (a) Original image for the decryption process, (b) The decrypted image

systems," *Genetic and Evolutionary Computation*, pp. 1369–1381, 2003.

[21] B. Srisuchinwong, "Chaos in a fractional-order Jerk model using tanh nonlinearity," *Proceedings of the 2nd Chaotic Modeling and Simulation International Conference*, pp. 1–8, 2009.

[22] F. Seredynski, P. Bouvry and Albert Y. Zomaya, "Cellular automata computations and secret key cryptography," *Parallel Computing*, vol. 30, no. 5, pp. 753–766, 2004.

[23] X. Tong and M. Cui, "Image encryption scheme based on 3D Baker with dynamical compound chaotic sequence cipher generator," *Signal Processing*, vol. 89, pp. 480–491, 2009.

[24] K. Wang, W. Pei, L. Zou, A. Song and Z. He, "On the security of 3D Cat map based symmetric image encryption scheme," *Physics Letters A*, vol. 343, no. 6, pp. 432–439, 2005.

[25] K. Wong, B. Kwok, and W. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2008.

[26] S. Wolfram, "Cryptography with cellular automata," *Advances in Cryptology*, LNCS 218, pp. 429–432, Springer, 1985.

[27] J. W. Yoon and H. Kim, "An image encryption scheme with a pseudorandom permutation based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 3998–4006, 2010.

Warakorn Srichavengsup obtained the B.Eng., M.Eng. and Ph.D. degree in Electrical Engineering from Chulalongkorn University, Bangkok, Thailand, in 1998, 2003 and 2009, respectively. He is currently a lecturer with the Department of Computer Engineering at Faculty of Engineering, Thai-Nichi Institute of Technology (TNI), Bangkok, Thailand. Prior to joining TNI, he was a visiting research student during 2008 with the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT). His main research interests are MAC protocol for high speed wireless local area networks, computer cryptography and information security.

Wimol San-Um was born in Nan Province, Thailand in 1981. He received B.Eng. Degree in Electrical Engineering and M.Sc. Degree in Telecommunications in 2003 and 2006, respectively, from Sirindhorn International Institute of Technology (SIIT), Thammasat University in Thailand. In 2007, he was a research student at

University of Applied Science Ravensburg-Weingarten in Germany. He received Ph.D. in mixed-signal very large-scaled integrated circuit designs in 2010 from the Department of Electronic and Photonic System Engineering, Kochi University of Technology (KUT) in Japan. He is currently with Master of Engineering Technology program, Faculty of Engineering, Thai-Nichi Institute of Technology (TNI). He is also the head of Intelligent Electronic Systems (IES) Research Laboratory. His areas of research interests are chaos theory, artificial neural networks, control automations, digital image processing, secure communications, and nonlinear dynamics of chaotic circuits and systems.

An Improved Online/Offline Identity-based Signature Scheme for WSNs

Ya Gao[1], Peng Zeng[1], Kim-Kwang Raymond Choo[2], and Fu Song[1]
(Corresponding author: Peng Zeng)

Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China[1]
Information Assurance Research Lab, University of South Australia, Adelaide SA, Australia[2]
(Email: pzeng@sei.ecnu.edu.cn)
(Received Aug. 9, 2015; revised and accepted Jan. 23, 2016)

Abstract

Online/offline signature schemes allow the signer to generate an online signature in real-time from a precomputed offline signature when presented with a document. Such schemes are particularly useful in resource-constrained wireless sensor network applications. In this paper, we describe an identity-based online/offline signature scheme based on bilinear maps, and prove the security of the scheme assuming the intractability of the Computational Diffie-Hellman Problem. More precisely, under the random oracle model, our scheme is proved to be secure against existential forgery on adaptively chosen message attack. As an extension to our scheme, we demonstrate how the scheme can be extended to allow a single user to sign multiple messages.

Keywords: Bilinear pairing, identity-based signature, online/offline signature, wireless sensor network

1 Introduction

With advances in sensor technologies in recent times, wireless sensor networks (WSNs) are increasingly popular in commercial, government and military settings (see [17, 24, 26, 31]). A WSN is a network of spatially distributed autonomous sensors deployed to monitor physical or environmental conditions, such as temperature and pressure. Sensor nodes cooperatively pass their data through the network to a main location. A WSN environment typically consists of a large number of resource-constrained sensor nodes and several control nodes (also known as base stations) [18]. Similar to Mobile Ad Hoc Networks [1, 2], the open nature of wireless communication result in WSNs being vulnerable to a wider range of attacks. Therefore, providing authentication for sensor data is of utmost importance in WSN applications [16, 27, 38, 39].

Since sensor nodes are typically resource constrained (e.g. in terms of memory and battery power), symmetric-key-based μTESLA-like schemes [12, 22, 23, 28] are more appropriate for actual deployment on the nodes due to their energy efficiency. However, these schemes are vulnerable to energy-depleting denial of service (DoS) attacks [3, 21]. Secret key distribution problem between senders and receivers is also a challenge when deploying WSNs [34]. In the last few years, several schemes based on public key cryptography [5, 9, 10, 14] have been proposed to provide real-time authentication and eliminate the key distribution/management problem, which reduces the protocol overhead. In a traditional public key infrastructure deployment, we would require a trusted certification authority to issue a certificate in order to authenticate the user's public key [11]. However, such an approach consumes substantial bandwidth and power due to the need for transmitting and verification of public key certificates [33, 34].

Shamir [32] introduced identity-based (ID-based) cryptosystems and signature schemes, which eliminate the need for checking the validity of certificates. A user can use his name, e-mail address or other identity attributes as the public key, and therefore, ID-based cryptography is a viable option for WSNs. For example, when a new node joins the network, other nodes do not need to keep the certificate in order to communicate in a secure and authenticated way. In order to further reduce the computational overhead of signature generation, online/offline technology is deployed in WSNs. An online/offline signature scheme was introduced by Even et al. [13], where the signing of a message is separated into two phases. The first phase is performed offline, which can be executed before the message to be signed is known. Upon receiving the message to be signed, the second phase is performed online, which utilizes the precomputation of the first phase. Activities that require significant computation resources, such as exponentiation, should be avoided in the online phase for efficiency. This property is useful in WSNs. The offline phase can be performed by the powerful base station, while the online phase can be executed by the sensor nodes [36, 37].

The first online/offline ID-based signature scheme is, perhaps, proposed by Xu et al. [35]. In the scheme, whenever a signature needs to be generated, the signer will execute the offline phase. In a WSN, when the offline phase is performed at the base station, the sensor nodes need to obtain the next offline signature from the base station whenever the node is generating a signature. This will resulted in increased communication overheads. It was subsequently discovered that Xu's scheme does not achieve the claimed security property [20]. In a separate work, Liu et al. [25] presented an online/offline ID-based signature scheme, which allows the signer to reuse the offline precomputed information in polynomial time. However, Kar [19] demonstrated that Liu et al.'s scheme does not include the case that randomly selects string contains all $0s$ or all $1s$ or the position of 1 in odd or even place. For these cases, the scheme would be vulnerable to malicious attacks. An improved scheme was then proposed. However, in this paper, we explain that the verification equation in Kar's scheme [19] is invalid, and it does not achieve the claimed security property (i.e. the signature is forgeable). As an illustration, we will show how to forge a signature in Kar's scheme. We also propose an improved scheme, with an accompanying security analysis.

The rest of the paper is organized as follows. In Section 2, we review the relevant definitions and outline the framework of the online/offline ID-based signature scheme. In Section 3, we revisit Kar's scheme and reveal a previously unpublished vulnerability by demonstrating how a signature can be forged. Our scheme is described in Section 4, followed by the security proof. We extend the basic scheme to an aggregate signature scheme in Section 5, before concluding the paper in Section 6.

2 Preliminaries

2.1 Bilinear Pairings

Definition 1. *Let k be a security parameter and q be a k-bit prime number. Let $\mathbb{G}_1$ denote a cyclic additive group of prime order q and $\mathbb{G}_2$ a cyclic multiplicative group with the same order. We assume that the discrete logarithm problem is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a map with the following properties:*

1) *Bilinearity: For any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.*
2) *Non-degeneracy: There exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$. Therefore, when P is a generator of $\mathbb{G}_1$, $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$.*
3) *Computability: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.*

The above bilinear map is also known as a bilinear pairing. The map $\hat{e}$ can be derived from either Weil pairing or Tate pairing on an elliptic curve over a finite field, and we refer the reader to [4, 6, 8, 15] for a more comprehensive description.

2.2 Mathematical Assumption

Let $\mathbb{G}$ be an abelian group of prime order q and P a generator of $\mathbb{G}$. We describe the following three mathematical problems in the additive group $(\mathbb{G}, +)$.

Discrete Logarithm Problem (DLP): Given $P, Q \in \mathbb{G}$, find an integer $n \in \mathbb{Z}_q^*$, such that $Q = nP$ whenever such an integer exists.

Decision Diffie-Hellman Problem (DDHP): For $a, b, c \in \mathbb{Z}_q^*$, given $P, aP, bP, cP \in \mathbb{G}$, decide whether $c \equiv ab \mod q$.

Computational Diffie-Hellman Problem (CDHP): For $a, b \in \mathbb{Z}_q^*$, given $P, aP, bP \in \mathbb{G}$, compute abP.

CDHP assumption: There exists no algorithm running in polynomial time, which can solve the CDHP problem with non-negligible probability.

(t', ε')-CDH group: A probabilistic algorithm $\mathcal{A}$ is said (t', ε')-break the CDHP in $\mathbb{G}$ if $\mathcal{A}$ runs at most time t', computes the CDHP with an advantage of at least ε'. We say that $\mathbb{G}$ is a (t', ε')-CDH group if no probabilistic algorithm $\mathcal{A}$ (t', ε')-breaks the CDHP in $\mathbb{G}$.

2.3 Framework

Definition 2. *The online/offline ID-based signature scheme comprises five polynomial time algorithms, namely: Setup, Extract, Offline Sign, Online Sign, Verify.*

Setup. *The master key and parameter generation algorithm is a probabilistic algorithm. On input a security parameter 1^k, the algorithm will output a master key msk and a parameter list $params$.*

Extract. *The signing key issuing algorithm is a deterministic algorithm. On input a user's identity id and a master key msk, the algorithm will return a pair of matching public and secret keys (pk_{id}, sk_{id}).*

Offline Sign. *The offline signing algorithm is a probabilistic algorithm. On input a parameter list $params$, the algorithm will return the generated offline signature σ_{off}.*

Online Sign. *The online signing algorithm is a probabilistic algorithm. On input a parameter list $params$, an identity id, a message m, and an offline signature σ_{off}, the algorithm will return the generated signature σ.*

Verify. *The verification algorithm is a deterministic algorithm. On input a parameter list $params$, an identity id, a message m, and a signature σ, the algorithm will return $'accept'$ if σ is valid and $'reject'$ otherwise.*

3 Revisiting Kar's Online/Offline ID-based Signature Scheme

3.1 Kar's Scheme

Kar's Scheme [19] comprises the following five polynomial time algorithms.

Setup. Given security parameters k, the Private Key Generator (PKG) chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ both of prime order q, a generator P of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and two collision resistant hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. Next, PKG will choose a master-key $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$. The system public parameters are given by $\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2)$.

Extract. Given an identity $ID \in \{0,1\}^*$, the secret key will be $d_{ID} = s \cdot Q_{ID}$, where $Q_{ID} = H_1(ID)$.

Offline Sign. At the offline stage, the signer computes

$$\hat{\alpha}_i = \hat{e}(P, P_{pub})^{2^i}, \ \forall \ i = 0, 1, \ldots, |q| - 1.$$

Online Sign. During this phase, the signer randomly selects $\beta \in \mathbb{Z}_q^*$ and computes two index sets $\mathcal{D} = \{1 \leq i \leq |q| \mid \beta[i] = 1\}$ and $\mathcal{C} = \{1 \leq i \leq |q| \mid \beta[i] = 0\}$, where $\beta[i]$ is the i^{th} bit of β. Next, the signer will compute $\psi_1 = \prod_{i \in \mathcal{D}} \hat{\alpha}_{i-1}$, $\psi_2 = \prod_{i \in \mathcal{C}} \hat{\alpha}_{i-1}$ and $\alpha = \psi_1 \psi_2$. Then, the signer randomly selects $\gamma \in \mathbb{Z}_q^*$ and computes $U = \gamma \cdot P$, $r = H_2(ID, U||m)$, and $V = (\gamma + \beta) \cdot P_{pub} + r d_{ID}$. The signature is $\sigma = (\alpha, U, V)$.

Verify. The signature is valid only if the following equation holds:

$$\hat{e}(V, P) \stackrel{?}{=} \alpha \cdot \hat{e}(Q_{ID}, P_{pub})^r \cdot \hat{e}(U, P_{pub}). \quad (1)$$

3.2 Previously Unpublished Vulnerabilities

We will now show that Equation (1) does not hold for general cases, even in the event that (α, U, V) is a valid signature for the message m and the identity ID. First we have

$$\begin{aligned}
\hat{e}(V, P) &= \hat{e}((\gamma + \beta)P_{pub} + r d_{ID}, P) \\
&= \hat{e}((\gamma + \beta)P_{pub}, P) \cdot \hat{e}(r d_{ID}, P) \\
&= \hat{e}(P_{pub}, (\gamma + \beta)P) \cdot \hat{e}(rsQ_{ID}, P) \\
&= \hat{e}(P_{pub}, \gamma P) \cdot \hat{e}(P_{pub}, \beta P) \cdot \hat{e}(rQ_{ID}, sP) \\
&= \hat{e}(P_{pub}, U) \cdot \hat{e}(P_{pub}, P)^{\beta} \cdot \hat{e}(rQ_{ID}, P_{pub}) \\
&= \hat{e}(P_{pub}, P)^{\beta} \cdot \hat{e}(Q_{ID}, P_{pub})^r \cdot \hat{e}(U, P_{pub}).
\end{aligned}$$

Thus, Equation (1) holds if, and only if, $\alpha = \hat{e}(P_{pub}, P)^{\beta}$. However, we have

$$\begin{aligned}
\alpha &= \psi_1 \psi_2 \\
&= (\prod_{i \in \mathcal{D}} \hat{\alpha}_{i-1})(\prod_{i \in \mathcal{C}} \hat{\alpha}_{i-1}) \\
&= \hat{\alpha}_0 \hat{\alpha}_1 \cdots \hat{\alpha}_{|q|-1} \\
&= \hat{e}(P, P_{pub})^{2^0} \hat{e}(P, P_{pub})^{2^1} \cdots \hat{e}(P, P_{pub})^{2^{|q|-1}} \\
&= \hat{e}(P, P_{pub})^{2^0 + 2^1 + \cdots + 2^{|q|-1}} \\
&= \hat{e}(P, P_{pub})^{2^{|q|} - 1}.
\end{aligned}$$

Since β is randomly selected from $\mathbb{Z}_q^*$, it is clear that $\beta \neq 2^{|q|} - 1 \mod q$ in general, which results in

$$\alpha \neq \hat{e}(P_{pub}, P)^{\beta};$$

thus, Equation (1) does not hold.

In addition to the above design flaw, we will show that the scheme is vulnerable to an existential forgery attack, in violation of their security claim. We reasonably assume that $\mathcal{A}$ is an attacker who has the public parameters

$$\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2).$$

$\mathcal{A}$ can execute the following steps to forge a signature $\sigma' = (\alpha', U', V')$ for a message m' and a legitimate identity ID.

1) $\mathcal{A}$ selects $U' \in \mathbb{G}_1$ and computes
$$r' = H_2(ID, U'||m').$$
2) $\mathcal{A}$ selects $V' \in \mathbb{G}_1$ and computes
$$\alpha' = \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q-1},$$
where $Q_{ID} = H_1(ID)$.
3) $\mathcal{A}$ sends the forgery signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID to the verifier.

When the verifier receives the signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID, the verifier will compute $r' = H_2(ID, U'||m')$ and check whether Equation (2) holds.

$$\hat{e}(V', P) \stackrel{?}{=} \alpha' \cdot \hat{e}(Q_{ID}, P_{pub})^{r'} \cdot \hat{e}(U', P_{pub}) \quad (2)$$

We now obtain:

$$\begin{aligned}
& \alpha' \cdot \hat{e}(Q_{ID}, P_{pub})^{r'} \cdot \hat{e}(U', P_{pub}) \\
= \ & \alpha' \cdot \hat{e}(r' Q_{ID}, P_{pub}) \cdot \hat{e}(U', P_{pub}) \\
= \ & \alpha' \cdot \hat{e}(r' Q_{ID} + U', P_{pub}) \\
= \ & \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q-1} \cdot \hat{e}(r' Q_{ID} + U', P_{pub}) \\
= \ & \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q} \\
= \ & \hat{e}(V', P)
\end{aligned}$$

The above equation holds because the group $\mathbb{G}_2$ has prime order q. Therefore, the forgery signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID will always be successfully verified. In other words, it is trivial to forge a signature. Consequently, this violates the claim by Kar that the scheme is secure against existential forgery on chosen message attack.

4 Our Proposed Signature Scheme

4.1 The Basic Scheme

In this section, we propose an improved online/offline signature scheme whose security is based on the assumption that CDHP is hard to solve. Our scheme contains the following five polynomial time algorithms.

Setup. Given a security parameter $k \in \mathbb{Z}$, this algorithm works as follows:

1) Generates a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Chooses a generater P of $\mathbb{G}_1$.
2) Selects a random $s \in \mathbb{Z}_q^*$ as the master key, and sets $P_{pub} = sP$.
3) Chooses two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$, which will be viewed as random oracles in our security proof. The system parameters are

$$Params = \{\mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, q, \hat{e}, H_1, H_2\}.$$

Extract. For a given identity $ID \in \{0,1\}^*$, the algorithm computes the associated private key $S_{ID} = s \cdot H_1(ID)$, where $Q_{ID} = H_1(ID)$ plays the role of the associated public key.

Offline Sign. During the offline stage, the signer computes:

$$Y_i = \hat{e}(P_{pub}, P)^{2^i},\ i = 0, 1, \ldots, \ell,$$

where $\ell = |q| - 1$.

Online Sign. During the online stage, given a private key S_{ID} and a message $m \in \{0,1\}^*$, the signer computes the followings:

1) Randomly chooses a number $y \in \mathbb{Z}_q^*$ and computes
$$Y = \prod_{0 \le i \le \ell} Y_i^{y[i]},$$
where $y[i]$ denotes the i^{th} bit of y, $0 \le i \le \ell$.
2) Randomly chooses $x \in \mathbb{Z}_q^*$, and computes $R = xP$ and
$$h = H_2(m, R, Y).$$
3) Computes $Z = (x + y)P_{pub} + hS_{ID}$.

The signature is $\sigma = (Y, R, Z)$.

Verify. In order to verify the signature σ of a message m for an identity ID, the verifier computes the followings:

1) Computes $h = H_2(m, R, Y)$.
2) Verifies whether the following equation holds.

$$\hat{e}(Z, P) \stackrel{?}{=} Y \cdot \hat{e}(R + hQ_{ID}, P_{pub}) \qquad (3)$$

Accepts if the above verification returns true, and rejects otherwise.

Consistency. Let $\sigma = (Y, R, Z)$ be a valid signature of a message m for an identity ID (in the case $Z = (x + y)P_{pub} + hS_{ID}$, $R = xP$, $h = H_2(m, R, Y)$, and $Y = \prod_{0 \le i \le \ell} Y_i^{y[i]}$), we have

$$\begin{aligned}
\hat{e}(Z, P) &= \hat{e}((x + y)P_{pub} + hS_{ID}, P) \\
&= \hat{e}((x + y)P_{pub}, P) \cdot \hat{e}(hS_{ID}, P) \\
&= \hat{e}(P_{pub}, (x + y)P) \cdot \hat{e}(hsQ_{ID}, P) \\
&= \hat{e}(P_{pub}, xP) \cdot \hat{e}(P_{pub}, yP) \cdot \hat{e}(hQ_{ID}, sP) \\
&= \hat{e}(P_{pub}, R) \cdot \hat{e}(P_{pub}, P)^y \cdot \hat{e}(hQ_{ID}, P_{pub}) \\
&= \hat{e}(P_{pub}, P)^y \cdot \hat{e}(R + hQ_{ID}, P_{pub})
\end{aligned}$$

Then, Equation (3) holds if and only if $Y = \hat{e}(P_{pub}, P)^y$. On the other hand,

$$\begin{aligned}
Y &= \prod_{0 \le i \le \ell} Y_i^{y[i]} \\
&= Y_0^{y[0]} Y_1^{y[1]} \ldots Y_\ell^{y[\ell]} \\
&= \hat{e}(P_{pub}, P)^{y[0]2^0} \cdot \hat{e}(P_{pub}, P)^{y[1]2^1} \cdots \hat{e}(P_{pub}, P)^{y[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y[0]2^0 + y[1]2^1 + \cdots + y[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^y.
\end{aligned}$$

Thus, we show the consistency of our signature scheme. In the next section, we will prove that our signature scheme is secure against existential forgery on adaptively chosen message attack under the CDHP assumption.

4.2 Security Proof

Let $\mathcal{S} = (\textbf{Setup}, \textbf{Extract}, \textbf{Offline Sign}, \textbf{Online Sign}, \textbf{Verify})$ denotes an online/offline ID-based signature scheme. We consider the following game, denoted by $\text{Game}_{\mathcal{S},\mathcal{A}}^{\text{EUF-ACM}}$, involving a probabilistic polynomial time algorithm $\mathcal{A}$:

1) The challenger, denoted by $\mathcal{F}$, runs the **Setup** algorithm to generate the system parameters *Params* and sends them to $\mathcal{A}$.
2) $\mathcal{A}$ performs the following queries as he wants:
   - Hash function query. $\mathcal{F}$ computes the value of the hash function for the requested input and sends the value to $\mathcal{A}$.
   - Extract query. When $\mathcal{A}$ produces an identity id, $\mathcal{F}$ will return the private key sk_{id} corresponding to id, which is obtained by running **Extract**.

- Sign query. Proceeding adaptively, $\mathcal{A}$ requests signatures on at most q_s messages of his choice $m_1, \ldots, m_{q_s} \in \{0,1\}^*$. $\mathcal{F}$ responds to each query with a signature σ_i $(1 \leq i \leq q_s)$, which is obtained by running **Offline Sign** and **Online Sign**.

3) After a polynomial number of queries, the adversary $\mathcal{A}$ produces a tuple (id^*, m^*, σ^*) whose secret key was not asked in any Extract queries and the pair (id^*, m^*) was not asked in any Sign queries. $\mathcal{A}$ wins the game if σ^* is a valid signature of m^* for id^*.

Definition 3. *An adversary $\mathcal{A}$ $(t, q_h, q_e, q_s, \varepsilon)$-breaks an online-offline ID-based signature scheme $\mathcal{S}$ if $\mathcal{A}$ wins the game $Game^{EUF\text{-}ACM}_{\mathcal{S},\mathcal{A}}$ with a non-negligible advantage (i.e. advantage of at least ε), running time at most t, and Hash functions, Extract and Sign queries at most q_h, q_e, q_s times, respectively. $\mathcal{S}$ is considered $(t, q_h, q_e, q_s, \varepsilon)$-existentially unforgeable under adaptively chosen message attacks if no adversary $(t, q_h, q_e, q_s, \varepsilon)$-breaks $\mathcal{S}$.*

We now prove the following lemma using the technique used in the BLS scheme [8].

Lemma 1. *Let $\mathbb{G}_1$ be an additive group and $\mathbb{G}_2$ a multiplicative group, which are two (t', ε')-CDH cyclic groups of the same prime order q. Let $\hat{e}$ be a computable bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. In the random oracle model, the proposed signature scheme is $(t, q_h, q_e, q_s, \varepsilon)$-secure against existential forgery under an adaptive chosen-message attack, in which t and ε satisfy*

$$\varepsilon \geq e(1+q_e)\varepsilon', \quad t \leq t' - (q_h + 2q_e + 2q_s)t_m.$$

Here, we denote by e the base of the natural logarithm and t_m the time for computing scalar multiplication. Let q_h, q_e, q_s respectively denote the number of H_1 queries, extract query and sign query, which the adversary is allowed to make.

Proof. Suppose that $\mathcal{A}$ is a forgery algorithm who $(t, q_h, q_e, q_s, \varepsilon)$-breaks the signature scheme and outputs a valid forged signature. The algorithm $\mathcal{B}$ simulates the challenger and interacts with the forgery algorithm $\mathcal{A}$. We can then use $\mathcal{A}$ to construct a t'-time algorithm $\mathcal{B}$ and solve the CDH problem with probability of at least ε'. Let P be a generator of $\mathbb{G}_1$. We now describe algorithm $\mathcal{B}$, which computes $abP \in \mathbb{G}_1$ for a randomly given CDH instance (P, aP, bP) where $a, b \in \mathbb{Z}_q^*$.

Setup. Algorithm $\mathcal{B}$ sets $P_{pub} = aP$ as the public key, and algorithm $\mathcal{A}$ obtains the system parameters $\{P, P_{pub}\}$ from $\mathcal{B}$.

H_1-query. Algorithm $\mathcal{A}$ is allowed to query the random oracle H_1 at any time. In order to respond to these queries, algorithm $\mathcal{B}$ maintains a list of tuples $\langle ID_j, \alpha_j, \beta_j, c_j \rangle$ denoted as L_1, which is initially empty. When $\mathcal{A}$ queries the oracle H_1 at a point $ID_i \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

1) If the query ID_i already appears on the H_1-list in a tuple $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$, algorithm $\mathcal{B}$ will respond with $H_1(ID_i) = \beta_i$. Otherwise, algorithm $\mathcal{B}$ generates a random coin $c_i \in \{0,1\}$, so that $Pr[c_i = 0] = 1/(1+q_e)$.
2) Algorithm $\mathcal{B}$ picks a random $\alpha_i \in \mathbb{Z}_q^*$ and computes $\beta_i = \alpha_i b^{1-c_i} P$.
3) Algorithm $\mathcal{B}$ adds the tuple $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ to the H_1-list and responds to $\mathcal{A}$ by setting $H_1(ID_i) = \beta_i$.

Extract query. Let ID_i be an extract query issued by $\mathcal{A}$. Algorithm $\mathcal{B}$ responds to this query as follows:

1) Algorithm $\mathcal{B}$ runs H_1-query to obtain $H_1(ID_i) = \beta_i$. Let $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list. If $c_i = 0$, then algorithm $\mathcal{B}$ reports failure and terminates.
2) Otherwise, we know $c_i = 1$; hence, $\beta_i = \alpha_i P$. Algorithm $\mathcal{B}$ computes $S_{ID_i} = \alpha_i \cdot P_{pub} = a \cdot (\alpha_i P)$ and responds to algorithm $\mathcal{A}$ with S_{ID_i}.

Sign query. Let m_i be a sign query issued by $\mathcal{A}$ with the identity ID_i, algorithm $\mathcal{B}$ responds to this query as follows:

1) Algorithm $\mathcal{B}$ runs the above algorithm for responding to H_1-query to obtain a β_i such that $H_1(ID_i) = \beta_i$. Let $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list.
2) Algorithm $\mathcal{B}$ randomly picks $x_i, y_i, h_i \in \mathbb{Z}_q^*$. Then, $\mathcal{B}$ computes $Y_i = \prod_{0 \leq k < |q|} Y_k^{y_i[k]}$ (where $Y_k = \hat{e}(P_{pub}, P)^{2^k}, \forall k = 0, 1, \ldots, |q|-1$), $R_i = x_i h_i P - y_i P - \beta_i h_i$, and $Z_i = x_i h_i P_{pub}$.
3) Algorithm $\mathcal{B}$ responds to algorithm $\mathcal{A}$ with $\sigma_i = (Y_i, R_i, Z_i)$.

We also remark that σ_i is always a valid signature on the message m_i for the identity ID_i.

$$\begin{aligned} & Y \cdot \hat{e}(R + hQ_{ID}, P_{pub}) \\ = \; & \hat{e}(P_{pub}, P)^y \hat{e}(R + hQ_{ID}, P_{pub}) \\ = \; & \hat{e}(yP + R + h\beta, P_{pub}) \\ = \; & \hat{e}(xhP, P_{pub}) \\ = \; & \hat{e}(xhP_{pub}, P) \\ = \; & \hat{e}(Z, P). \end{aligned}$$

We apply the oracle replay attack (coined by Pointcheval and Stern [29, 30]). In such an attack, we need to pay attention to the problem of collisions of query results as mentioned in proof of Lemma 4 in [29]. If no collision occurs, algorithm $\mathcal{A}$ outputs a valid (ID^*, m^*, σ^*) such that the pair (ID^*, m^*) was not asked in any Sign queries. If there is no tuple on the H_1-list containing ID^*, algorithm $\mathcal{B}$ will issue such a query for $H_1(ID^*)$ to ensure that the tuple exists.

Similar to the *forking lemma* [29], by replaying $\mathcal{B}$ using the same random tape but different choices of H_1, we obtain signatures (ID, m, h, Y, R, Z) and (ID, m, h', Y, R, Z') which are valid with respect to the hash functions H_1 and H_1' with different values $h \neq h'$ on (m, Y, R), respectively.

Algorithm $\mathcal{B}$ obtains the corresponding tuple from the L_1-list. If $c = 1$, algorithm $\mathcal{B}$ outputs failure and terminates. Otherwise, we know $c = 0$; thus, $H_1(ID) = \beta = b\alpha P$. Algorithm $\mathcal{B}$ computes $Z - Z' = (h - h')S_{ID} = (h - h')sQ_{ID} = (h - h')ab\alpha P$ and $abP = (Z - Z')(h - h')^{-1}/\alpha$, where abP is the solution to the CDH instance (P, aP, bP).

We will now show that algorithm $\mathcal{B}$ solves the given CDH instance (P, aP, bP) with probability at least ε'. We analyze the three events required for algorithm $\mathcal{B}$ to succeed:

- ε_1 : Algorithm $\mathcal{B}$ does not abort as a result of any Extract queries of algorithm $\mathcal{A}$.
- ε_2 : Algorithm $\mathcal{A}$ generates a valid message-signature forgery (Y, R, Z).
- ε_3 : The event ε_2 occurs and $c = 0$ for tuples containing ID on the L_1-list.

Algorithm $\mathcal{B}$ succeeds if all these events happen, and the corresponding probability is

$$Pr[\varepsilon_1 \wedge \varepsilon_3] = Pr[\varepsilon_1] \cdot Pr[\varepsilon_2|\varepsilon_1] \cdot Pr[\varepsilon_3|\varepsilon_1 \wedge \varepsilon_2] \quad (4)$$

□

Claim 1. *The probability that algorithm $\mathcal{B}$ does not abort as a result of any Extract queries asked by algorithm $\mathcal{A}$ is at least $(1 - 1/(1 + q_e))^{q_e}$.*

Proof. We assume that $\mathcal{A}$ does not query the signature of the same message twice. The probability that algorithm $\mathcal{B}$ does not abort is at least $(1 - 1/(1 + q_e))^i$ after i $(0 \leq i \leq q_e)$ signature queries were asked by algorithm $\mathcal{A}$. It is clear that the claim is true when $i = 0$. Let ID_i be the i-th extract query asked by $\mathcal{A}$, and $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list. Before issuing the extract query, only $H_1(ID_i) = \beta_i$ depends on the random coin c_i, and distribution on $H_1(ID_i)$ is the same as c_i's. Thus, the probability that the Extract query causes $\mathcal{B}$ to abort is at most $1/(1 + q_e)$. Based on the inductive hypothesis and the independence of c_i, the probability that $\mathcal{B}$ does not abort after this signature query is at least $(1 - 1/(1 + q_e))^i$. This proves the claim; as $\mathcal{A}$ makes at most q_e extract queries, the probability that $\mathcal{B}$ does not abort is at least $(1 - 1/(1 + q_e))^{q_e} \geq 1/e$. □

Claim 2. *If $\mathcal{B}$ does not abort as a result of any extract queries of algorithm $\mathcal{A}$, then $\mathcal{A}$'s view is identical to its view in the real attack. Hence, $Pr[\varepsilon_2|\varepsilon_1] \geq \varepsilon$.*

Proof. As h_1 and h_2 are two collision resistant hash functions, responses to h_1-queries and h_2-queries are similar to those in a real attack. All responds to the Extract queries and signature queries are valid. Therefore, $\mathcal{A}$ generates a valid message-signature pair with probability of at least ε. Hence, $Pr[\varepsilon_2|\varepsilon_1] \geq \varepsilon$. □

Claim 3. *The probability that algorithm $\mathcal{B}$ does not abort after $\mathcal{A}$ outputs a valid forgery is at least $1/(1 + q_e)$. Hence, $Pr[\varepsilon_3|\varepsilon_1 \wedge \varepsilon_2] = 1/(1 + q_e)$.*

Proof. Suppose that events ε_1 and ε_2 occurred, algorithm $\mathcal{B}$ will abort only when $\mathcal{A}$ outputs a forgery message-signature pair (m, σ) and $c = 0$ in the tuple $\langle ID, \alpha, \beta, c \rangle$ on the h_1-list. If $\mathcal{A}$ did not issue an Extract query for m_i, only $H_1(ID_i)$ depends on the random coin c_i, and distribution on $H_1(ID_i)$ is the same as c_i's. Due to that $\mathcal{A}$ could not issue an extract query for m, c is independent of $\mathcal{A}$'s current view. Therefore, $Pr[c = 0|\varepsilon_1 \wedge \varepsilon_2] = 1/(1 + q_e)$. □

According to Equation (4) and using the bounds from the above claims, algorithm $\mathcal{B}$ succeeds with probability at least $1/e \cdot \varepsilon \cdot 1/(1 + q_e)$.

If algorithm $\mathcal{A}$ takes time t to run, algorithm $\mathcal{B}$ takes time t and with the time required to respond to $(q_h + q_e + q_s)$ H_1-queries, q_e extract queries, and q_s signature queries. Each hash query and extract query require one scalar multiplication in $\mathbb{G}_1$, and each signature query requires four scalar multiplications in $\mathbb{G}_1$. We assume that one scalar multiplication in $\mathbb{G}_1$ takes time t_m. Thus, algorithm $\mathcal{B}$ takes time of at most $t + (q_h + 2q_e + 2q_s)t_m$.

Lemma 2. *(Lemma 4 in [29]) If there is an algorithm $\mathcal{A}_0$ for an adaptively chosen message attack against our scheme which queries H_1, Extract, and Sign at most q_h, q_e, q_s times respectively, and has running time t_0 and advantage $\varepsilon_0 \geq 10(1 + q_s)(q_h + q_s)/2^k$, then CDHP can be solved with probability $\varepsilon' \geq 1/9$ with run time of $t' \leq 23q_h t_0/\varepsilon_0$.*

Combining the above lemmas, we have the following theorem.

Theorem 1. *If there is an algorithm $\mathcal{A}$ for an adaptively chosen message attack to our scheme which queries H_1, Extract, and Sign at most q_h, q_e, q_s times respectively, and has running time t and advantage $\varepsilon \geq 10e(1 + q_e)(1 + q_s)(q_h + q_s)/2^k$, then CDHP can be solved with probability $\varepsilon' \geq 1/9$ within running time $t' \leq 23q_h(t + (q_h + 2q_e + 2q_s)t_m)/(e(1 + q_e))$.*

5 The Extended (Aggregation) Scheme

If a sensor node is able to sign multiple messages (for example n messages) and the size of the resulting signature is smaller than n times the size of a single signature, such an aggregated signature is practical for WSN deployment due to the reduced communication overheads. We propose the following aggregation signature as an extension to our online/offline signature scheme.

Setup. Given a security parameter $k \in \mathbb{Z}$, this algorithm works as follows:

1) Generates a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Chooses a generater P in $\mathbb{G}_1$.
2) Selects a random $s \in \mathbb{Z}_q^*$ as the master key, and sets $P_{pub} = sP$.
3) Chooses two collision resistant hash function $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. The system parameters are

$$Params = \{\mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, q, \hat{e}, H_1, H_2\}.$$

Extract. For a given identity $ID \in \{0,1\}^*$, compute $Q_{ID} = H_1(ID)$ and set $S_{ID} = sQ_{ID}$ as a private key of ID.

Offline Sign. At the offline stage, the signer computes:

$$Y_i = \hat{e}(P_{pub}, P)^{2^i},\ i = 0, 1, \ldots, \ell,$$

where $\ell = |q| - 1$.

Online Sign. During the online stage, given a private key S_{ID} and n messages $m_j \in \{0,1\}^*$,$1 \le j \le n$, the signer computes the followings:

1) For any $1 \le j \le n$, randomly chooses $y_j \in \mathbb{Z}_q^*$ and computes

$$Y^{(j)} = \prod_{0 \le i \le \ell} Y_i^{y_j[i]},$$

where $y_j[i]$ denotes the i^{th} bit of y_j.
2) For any $1 \le j \le n$, randomly chooses $x_j \in \mathbb{Z}_q^*$, and computes

$$h_j = H_2(m_j, R_j, Y^{(j)}),$$

where $R_j = x_j P$.
3) Computes $Z_j = (x_j + y_j)P_{pub} + h_j S_{ID},\ 1 \le j \le n$ and $Z = \sum_{j=1}^{n} Z_j$.

The signature is

$$\sigma = (Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}, R_1, R_2, \ldots, R_n, Z).$$

Verify. In order to verify the signature $\sigma = (Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}, R_1, R_2, \ldots, R_n, Z)$ for the n messages $m_j,\ j = 1, 2, \ldots, n$, and the identity ID, the verifier computes the followings:

1) Computes $h_j = H_2(m_j, R_j, Y^{(j)}),\ j = 1, 2, \ldots, n$.
2) Verifies whether the following equation holds

$$\hat{e}(Z, P) \stackrel{?}{=} \prod_{j=1}^{n} (Y^{(j)} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub})) \quad (5)$$

Accepts if it is equal, and rejects otherwise.

Consistency. Let $\sigma = (Y^{(n)}, R_n, Z)$ be a valid signature for n messages $m_j,\ j = 1, 2, \ldots, n$, and identity ID, we have

$$\begin{aligned}
& \hat{e}(Z, P) \\
= \ & \hat{e}(\sum_{j=1}^{n} Z_j, P) \\
= \ & \hat{e}(\sum_{j=1}^{n} ((x_j + y_j)P_{pub} + h_j S_{ID}), P) \\
= \ & \prod_{j=1}^{n} \hat{e}((x_j + y_j)P_{pub} + h_j S_{ID}, P) \\
= \ & \prod_{j=1}^{n} (\hat{e}((x_j + y_j)P_{pub}, P) \cdot \hat{e}(h_j S_{ID}, P)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, (x_j + y_j)P) \cdot \hat{e}(h_j s Q_{ID}, P)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, x_j P) \cdot \hat{e}(P_{pub}, y_j P) \cdot \hat{e}(h_j Q_{ID}, sP)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, R_j) \cdot \hat{e}(P_{pub}, P)^{y_j} \cdot \hat{e}(h_j Q_{ID}, P_{pub})) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, P)^{y_j} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub}))
\end{aligned}$$

Then, Equation (5) holds if and only if

$$Y^{(j)} = \hat{e}(P_{pub}, P)^{y_j},\ j = 1, 2, \ldots, n.$$

On the other hand, for any $1 \le j \le n$, we have

$$\begin{aligned}
Y^{(j)} &= \prod_{0 \le i \le \ell} Y_i^{y_j[i]} \\
&= Y_0^{y_j[0]} Y_1^{y_j[1]} \ldots Y_\ell^{y_j[\ell]} \\
&= \hat{e}(P_{pub}, P)^{y_j[0]2^0} \ldots \hat{e}(P_{pub}, P)^{y_j[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y_j[0]2^0 + \cdots + y_j[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y_j}.
\end{aligned}$$

Thus, $\hat{e}(Z, P) = \prod_{j=1}^{n} (Y^{(j)} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub}))$ and the verification is successful.

We refer to [7] for a detailed description of the security model and the security proof. Under the random oracle model, our aggregation signature scheme is also secure against existential forgery on adaptively chosen message attack.

6 Conclusion

In this paper, we proposed an online/offline ID-based signature scheme and proved that the scheme is secure against existential forgery on adaptively chosen message

attack in random oracle model, under the assumption that CDHP is intractable. We also extended the basic scheme to provide the ability for a user to sign multiple messages.

Acknowledgments

This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (Grant No. U1509219), the National Natural Science Foundation of China (Grant Nos. 61402179, 61321064 and 61103222), the Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110076120016), and the Pujiang Talent Project of the Shanghai Science and Technology Committee (Grant No. 14PJ1403200).

References

[1] A. Aburumman, K. K. R. Choo, "A domain-based multi-cluster SIP solution for mobile ad hoc network," in *International Conference on Security and Privacy in Communication Networks*, pp. 267–281, 2014.

[2] A. Aburumman, W. J. Seo, M. R. Islam, M. K. Khan, K. K. R. Choo, "A secure cross-domain SIP solution for mobile ad hoc network using dynamic clustering," in *International Conference on Security and Privacy in Communication Networks*, pp. 649–664,2015.

[3] A. Agah, S. K. Das, "Preventing doS attacks in wireless sensor networks: A repeated game theory approach," *International Journal of Network Security*, vol. 5, no. 2, pp. 145–153, 2007.

[4] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, et al. "Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology (Crypto'02),* Springer Berlin Heidelberg, pp. 354–369, 2002.

[5] C. Benzaid, K. Lounis, A. Al-Nemrat, et al. "Fast authentication in wireless sensor networks," *Future Generation Computer Systems,* vol. 55, pp. 362–375, 2016.

[6] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology (Crypto'01),* Springer Berlin Heidelberg, pp. 213–229, 2001.

[7] D. Boneh, C. Gentry, B. Lynn, et al. "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology (Eurocrypt'03),* Springer Berlin Heidelberg, pp. 416–432, 2003.

[8] D. Boneh, B. Lynn, H. Shacham, "Short signatures from the Weil pairing," in *Advances in Cryptology (Asiacrypt'01),* Springer Berlin Heidelberg, pp. 514–532, 2001.

[9] X. Cao, W. Kou, L. Dang, et al. "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Computer Communications,* vol. 31, no. 4, pp. 659–667, 2008.

[10] C. Y. Cheng, I. C. Lin, S. Y. Huang, "An RSA-like scheme for multiuser broadcast authentication in wireless sensor networks," *International Journal of Distributed Sensor Networks,* 2015.

[11] L. Cheng, Q. Wen, Z. Jin, et al. "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Information Sciences,* vol. 295, pp. 337–346, 2015.

[12] O. Delgado-Mohatar, A. Fúster-Sabater, J. M. Sierra, "A light-weight authentication scheme for wireless sensor networks," *Ad Hoc Networks,* vol. 9, no. 5, pp. 727–735, 2011.

[13] S. Even, O. Goldreich, S. Micali "On-line/off-line digital signatures," in *Advances in Cryptology (Crypto'89) Proceedings,* Springer New York, pp. 263–275, 1990.

[14] X. Fan, G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks,* vol. 10, no. 4, pp. 723–736, 2012.

[15] S. D. Galbraith, K. Harrison, D. Soldera, "Implementing the Tate pairing," *Algorithmic Number Theory,* Springer Berlin Heidelberg, pp. 324–337, 2002.

[16] Y. Gao, P. Zeng, K. K. R. Choo, "Multi–sender broadcast authentication in wireless sensor networks," in *IEEE Tenth International Conference on Computational Intelligence and Security,* pp. 633–637, 2014.

[17] M. Ge, K. K. R. Choo, H. Wu, Y. Yu, "Survey on key revocation mechanisms in wireless sensor networks," *Journal of Network and Computer Applications,* (In press), 2016.

[18] K. Grover, A. Lim, "A survey of broadcast authentication schemes for wireless networks," *Ad Hoc Networks,* vol. 24, pp. 288–316, 2015.

[19] J. Kar, "Provably secure online/off-line identity-based signature scheme for wireless sensor network," *International Journal of Network Security,* vol. 16, no. 1, pp. 29–39, 2014.

[20] F. Li, M. Shirase, T. Takagi, "On the security of online/offline signatures and multisignatures from acisp06," *Cryptology and Network Security,* Springer Berlin Heidelberg, pp. 108–119, 2008.

[21] W. T. Li, T. H. Feng, M. S. Hwang, "Distributed detecting node replication attacks in wireless sensor networks: A survey," *International Journal of Network Security,* VOL. 16, NO. 5, PP. 323–330, 2014.

[22] X. Li, N. Ruan, F. Wu, et al. "Efficient and enhanced broadcast authentication protocols based on multilevel μTESLA," in *IEEE International Performance Conference on Computing and Communications,* pp. 1–8, 2014.

[23] D. Liu, P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS),* vol. 3, no. 4, pp. 800–836, 2004.

[24] D. Liu, P. Ning, *Security for Wireless Sensor Networks*, Springer, 2007.

[25] J. K. Liu, J. Baek, J. Zhou, et al. "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security,* vol. 9, no. 4, pp. 287–296, 2010

[26] M. Luk, A. Perrig, B. Whillock, "Seven cardinal properties of sensor network broadcast authentication," in *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks,* pp. 147–156, 2006.

[27] J. Nam, K. K. R. Choo, M. Kim, J. Paik and D. Won, "Efficient and anonymous two-factor user authentication in wireless sensor networks: Achieving user anonymity with lightweight sensor computation," *PLOS ONE,* vol. 10, no. 4, pp. e0116709, 2015.

[28] A. Perrig, R. Szewczyk, J. D. Tygar, et al. "SPINS: Security protocols for sensor networks," *Wireless Networks,* vol. 8, no. 5, pp. 521–534, 2002.

[29] D. Pointcheval, J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology,* vol. 13, no. 3, pp. 361–396, 2000.

[30] D. Pointcheval, J. Stern, "Security proofs for signature schemes," in *Advances in Cryptology (Eurocrypt'96),* Springer Berlin Heidelberg, pp. 387–398, 1996.

[31] K. Ren, W. Lou, "Communication security in wireless sensor networks," *Worcester Polytechnic Institute,* 2007.

[32] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology,* Springer Berlin Heidelberg, pp. 47–53, 1985.

[33] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal of Network Security,* pp. 1–8, 2014.

[34] K. A. Shim, Y. R. Lee, C. M. Park, "EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks," *Ad Hoc Networks,* vol. 11, no. 1, pp. 182–189, 2013.

[35] S. Xu, Y. Mu, W. Susilo, "Online/offline signatures and multisignatures for AODV and DSR routing security," *Information Security and Privacy,* Springer Berlin Heidelberg, pp. 99–110, 2006.

[36] A. C. C. Yao, Y. Zhao, "Online/offline signatures for low-power devices," *IEEE Transactions on Information Forensics and Security,* vol. 8, no. 2, pp. 283–294, 2013.

[37] A. A. Yavuz, "An efficient real-time broadcast authentication scheme for command and control messages," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 10, pp. 1733–1742, 2014.

[38] P. Zeng, Z. Cao, K. K. R. Choo, S. Wang, "Security weakness in a dynamic program update protocol for wireless sensor networks," *IEEE Communications Letters,* vol. 13, no. 6, pp. 426–428, 2009.

[39] P. Zeng, K. K. R. Choo, D. Z. Sun, "On the security of an enhanced novel access control protocol for wireless sensor networks," *IEEE Transactions on Consumer Electronics,* vol. 56, no. 2, pp. 566–569, 2010.

Ya Gao received her B.S. degree in Information Security from Shanghai University of Electric Power in 2013. She is currently pursuing the M.S. degree with the Department of Cryptography and Network Security from East China Normal University. Her research interests include cryptography, information security, and wireless sensor network.

Peng Zeng received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University in 2009 and the M.S. degree in pure mathematics from East China Normal University in 2003, respectively. He is currently an associate professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. His research expertise include applied cryptography, network information security, and coding theory.

Kim-Kwang Raymond Choo received his Ph.D. degree from Queensland University of Technology in 2006. He is an associate professor at the University of South Australia. He has an interdisciplinary expertise in cyber security and digital forensics. He was named one of 10 Emerging Leaders in the Innovation category of The Weekend Australian Magazine / Microsoft's Next 100 series in 2009, and is the recipient of various awards including ESORICS 2015 Best Research Paper Award, Highly Commended Award from Australia New Zealand Policing Advisory Agency (ANZPAA), British Computer Society's Wilkes Award, Fulbright Scholarship, and 2008 Australia Day Achievement Medallion.

Fu Song is a lecturer at School of Computer Science and Software Engineering of East China Normal University, Peoples Republic of China. He received his Ph.D. in Computer Science from University Paris 7 in 2013 and M.Sc. from Software Engineering Institute of East China Normal University in 2009. His major research interests include software verification (e.g., infinite-state system modeling, temporal logics and model-checking), computer security (e.g., malware detection and binary code disassembly).

An Efficient and Robust Hybrid Watermarking Scheme for Text-images

Lamri Laouamer[1,2] and Omar Tayan[3]
(Corresponding author: Lamri Laouamer)
Department of Management Information Systems, CBE, Qassim University[1]
P.O. Box 6633 Buraidah, 51452 Qassim, KSA
Lab-STICC (UMR CNRS 6285), University of Bretagne Occidentale[2]
20 avenue Victor Le Gorgeu, BP817-CS 93837-29238 Brest Cedex, France
IT Research Center for the Holy Quran (NOOR) & College of Computer Science and Engineering, Taibah University[3]
Al-Madinah Al-Munawwarah 41411, KSA
(Email: laoamr@qu.edu.sa)
(Received Aug. 13, 2015; revised and accepted Nov. 27, 2015)

Abstract

Addressing fraud and illegal use of multimedia requires the development of more powerful and robust algorithms. One of the solutions that can contribute significantly to solve such problems can be found in the use of image watermarking. In image watermarking, a watermark is introduced within an image in order to protect it against illegal use. This paper proposes a new hybrid text-image watermarking algorithm based on the singular value decomposition (SVD) and the discrete cosine transform (DCT) with linear interpolation in the embedding/extraction process. Despite a considerable number of works found to-date, the robustness in existing watermark approaches remains a major challenge worthy of further improvement. Furthermore, text-images are used as samples in this work to test our scheme under further challenges and constraints imposed by such images on the embedding techniques used. In contrast to many existing approaches, we achieved through the proposed algorithm a high robustness results against the most dangerous attack scenarios. A major contribution in this paper is found in our unique watermark extraction scheme which differs from the existing literature and takes into account three inputs including the attacked watermarked image. Finally, we discuss the results obtained for our approach under various attack scenarios.

Keywords: Attacks, DCT, linear interpolation, robustness, SVD, watermark

1 Introduction

The evolution of communication technologies and data transmission has enhanced global access to information. Consequently, the dissemination and sharing of digital data has become easily accessible to users. However, the robustness issue has become an increasing problem since existing protection techniques relying solely on encryption have become insufficient to address the advancing requirements of data protection. The influence of digital-watermarking is increasing as a solution for countering various forms of illegal manipulation and piracy. Essentially, it consists of introducing an invisible signature in the host data, and then detecting possible manipulations applied on the watermarked data. Several techniques have been proposed in the literature, however, it remains that the invisibility versus robustness compromise under all attack scenarios remains an elusive goal in the research community.

In this paper, we focus on image watermarking and highlight the image watermarking approaches based on the hybrid singular value decomposition (SVD) and the discrete cosine transform (DCT) techniques. For this purpose, we expose some important image watermarking works from the literature based on SVD and DCT. Before examining those works, we review some important concepts of the SVD and the DCT transforms.

2 Background and State-of-art

The SVD Transform consists of factorizing a matrix M, into three matrices (components) U, S, V such that:

$$[M] = [U][S][V^T]$$

In fact, the inverse transform of the SVD is not entirely reversible, but rather is the product (Equation (1)):

$$M_{mn} = U_{mm}.S_{mn}.V_{nn}^T \quad (1)$$

where m, n are the image size (m represents the rows and n the columns). And U_{mm} is the left singular vector,

S_{mn} are the singular values and V_{nn} is the right singular vector. The DCT transform consists of changing the data from the spatial domain into the frequency domain. The corresponding DCT transform blocks are given by the following Equation (2):

$$F(u,v) = \frac{2}{N}c(u)c(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} pixel(x,y) \\ cos[\frac{\pi}{N}u(x+\frac{1}{2}]cos[\frac{\pi}{N}v(y+\frac{1}{2}] \quad (2)$$

The DCT inverse is given by the Equation (3):

$$f(x,y) = \frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} c(u)c(v)F(u,v) \\ cos[\frac{\pi}{N}u(x+\frac{1}{2}]cos[\frac{\pi}{N}v(y+\frac{1}{2}] \quad (3)$$

Where $c(u)$, $c(v) = (2)^{-1/2}$ for u, $v = 0$, $c(u)$, $c(v) = 1$ for u, $v = 1, 2, \cdots, N-1$, pixel(x, y) is the pixel value at position (x,y). The authors in [1] applied a differential evolution (DE) algorithm in the DCT domain to balance the trade-off between robustness and imperceptibility by exploring multiple scaling factors in image watermarking. The DC coefficients from each block were collected to construct a low-resolution approximation image and applied SVD on this image approximation. Experimental results show that the proposed scheme maintained a satisfactory image quality, with the watermark still identifiable following serious distortion to the image. In [2], an optimal DCT-SVD based image watermarking scheme using Pareto-based Multi-objective Genetic Algorithm (MOGA) was presented. After applying the DCT to the cover image, they map the DCT coefficients in a zigzag order into four quadrants, and apply SVD to each quadrant. The proposed algorithm in [2] was not perfectly robust, particularly against rotation attacks.

The work presented in [6] consists of a new robust hybrid image watermarking scheme based on SVD and DCT. After applying SVD to the cover image blocks, DCT was performed on the macro block comprised of the first singular values (SVs) of each image block. In the work by [6], an improved watermarking extraction scheme was demonstrated, particularly against median-filtering, rotation and cropping attacks. A robust lossless copyright protection scheme based on overlapping DCT and SVD was presented in [12] where direct current (DC) coefficients were extracted from the transformed blocks to form a DC-map. A series of random positions were selected on the map, and SVD was applied to construct an ownership share which is used for copyright verification. Experimental results were conducted to demonstrate the robustness of the proposed algorithm against several kinds of attacks, but with some weaknesses in the case of cropping, rotation and print-scan attacks.

The approach proposed in [4] presents a normalization-based robust image watermarking scheme which encompasses SVD and DCT techniques. The host image is first normalized into standard form and divided into non-overlapping image blocks. A watermark bit is then embedded in the high frequency band of an SVD-DCT block by imposing a particular relationship between two pseudo-randomly selected DCT coefficients. The experimental results show that the proposed approach was not perfectly robust against many attack scenarios.

In this paper, we propose a new hybrid algorithm based on SVD and DCT for the protection of online textual-images against several kinds of known attacks. In fact, the extraction method requires three inputs contrarily to the conventional semi-blind methods. The proposed embedding method is also based on linear interpolation with invisible watermarking. Additionally, text-images are applied as inputs since they present a particularly interesting research challenge (and relatively unexplored branch of image-watermarking) due to the limited solution-space available in the host-images used, as evidenced in the limited use of colours and textures, with clearly defined characters and whitespace etc. Nevertheless, our scheme can also be applied on general images comprising of a rich colour-set and more relaxed-constraints on the embedding process as considered in most related literature works. To determine the robustness of the proposed algorithm, we conducted several attacks on the watermarked images and compared the original watermark to the extracted one. The following section details our proposed algorithm.

3 Proposed Watermark Embedding Scheme

In this work, the embedding process is achieved using a linear interpolation of type (Equation (4)):

$$i_w = (1-\alpha)w + \alpha i \quad (4)$$

where i_w, w, i are the RGB images, the watermarked image, the original image, and the watermark respectively, and α, is a variable between 0 and 1 as explained in the Equation (5):

$$U_{iw} = U_i, S_{iw} = (1-\alpha)S_w + \alpha S_i \text{ and } V_{iw} = V_i \quad (5)$$

The illegible components U_{iw}, S_{iw}, V_{iw} are the corresponding matrices of the image i_w. Hence, obtaining the product of those three components (Equation (6)) gives a significant (readable) image i_w.

$$i_w = U_{iw} \times S_{iw}b \times V_{iw}^T \quad (6)$$

In our proposed embedding process, only the S-component (e.g. the singular values matrix) of the images i and w were used to achieve the data embedding (Equation (7)), which suggests that:

$$U_{iw} = U_i, \ \ S_{iw} = (1-\alpha)S_w + \alpha S_i \text{ and } V_{iw} = V_i \quad (7)$$

Obtaining an invisible watermark requires that the value of α (e.g. our used watermarking key) be set close

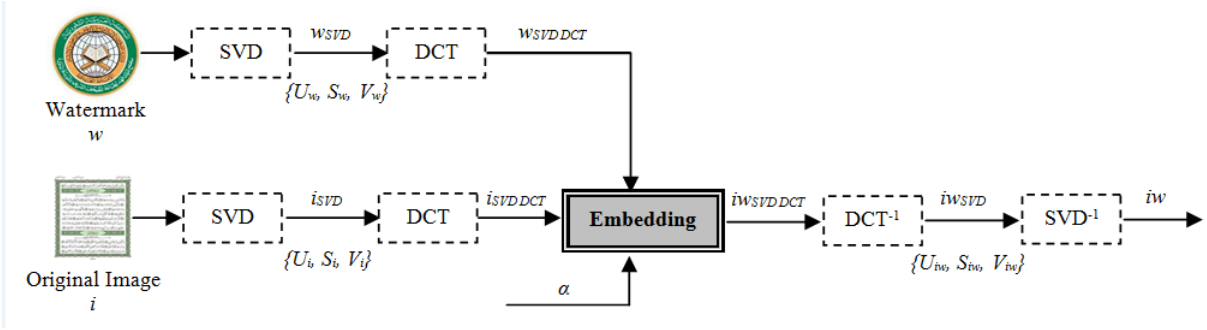


Figure 1: Watermark embedding algorithm

to 1 with $i_w \rightarrow i$, while a visible watermark signifies that α should be set close to 0 with $i_w \rightarrow w$. We summarize our embedding scheme in Figure 1.

Algorithm 1 Embedding algorithm

1: First compute the matrices: U, S, V , corresponding to both the original image i and the watermark w.
2: Apply the DCT embedding on the S_i and S_w components produced, suggesting that the embedding exists for only the singular-value matrices (e.g. following the DCT operation on the singular-value matrix of the original image i and the DCT operation on the singular-value matrix of the watermark). The embedding formula applied at this stage is given as follows: $U_{iw}=U_i$, DCT $(S_{iw})=(1-\alpha)DCT(S_w) + \alpha\ DCT(S_i)$ and $V_{iw}=V_i$
3: Next
4: Calculate the DCT inverse of the $DCT(S_{iw})$, and apply the SVD^{-1} term to obtain the watermarked image i_w as a result of the embedding. It is important to mention that the SVD^{-1} term does not mean the entire inverse transform, but rather, it is the product of the three matrices U, S, V^T. This suggests that SVD process is not completely reversible.

4 Proposed Watermark Extraction Scheme

In our proposed algorithm, the extraction process consists of applying three inputs i_w, w and i_{wa}, representing the watermarked image, the watermark, and the attacked watermarked-image respectively. This unique combination of inputs and operations in the extraction process was not found elsewhere in the related literature and was used effectively to obtain promising results through perfect extraction of the watermark as reported in the next section. The extraction process is as illustrated in Figure 2.

For the embedding operation (Equation (8)), we note that:

$$S_{wi} = (1-\alpha)DCT(S_{iw}) + \alpha DCT(S_w) \quad (8)$$

Next, the Unmark operation as illustrated in the Equation (9), consists of the reverse embedding process, which follows from:

$$W_{\alpha SVD} = \frac{1}{\alpha} W_{iSVD} - \frac{1-\alpha}{\alpha} i_{waSVD} \quad (9)$$

where w_a, i_{wa} are the extracted watermark and the attacked watermarked-image respectively. We note by Siw the singular value of the watermarked image i_w, however the S_{wi} consists to the results of the watermark embedding in the image i_w and not the image i (it is simply for differentiating notations).

Algorithm 2 Extraction algorithm

1: First, compute the matrices: U, S, V , for the watermark w, the watermark-image i_w, and the attacked watermarked image i_{wa}.
2: Apply the embedding of DCT(S_w) using the $DCT(Si_w)$ term. The embedding result of this step gives $w_{iSVDDCT}$. This embedding exists for only the singular-value matrices of the DCTs of S_w and S_{iw} respectively (e.g. following from the DCT operation on the singular-values matrix of the watermark w, and the DCT operation on the singular-values matrix of the watermarked image). The embedding formula used at this stage is given by (Equation (10)):

$$DCT(S_{wi}) = (1-\alpha)DCT(S_{iw}) + \alpha DCT(S_w) \quad (10)$$

3: Next, run the Unmark process between $i_{waSVDDCT}$ and $w_{iSVDDCT}$, which gives the matrix $w_{aSVDDCT}$ according to the Equation (11):

$$W_{\alpha SVDDCT} = \frac{1}{\alpha} W_{iSVDDCT} - \frac{1-\alpha}{\alpha} i_{waSVDDCT} \quad (11)$$

4: Finally, compute the DCT and the SVD inverse to obtain the extracted watermark, w_a.

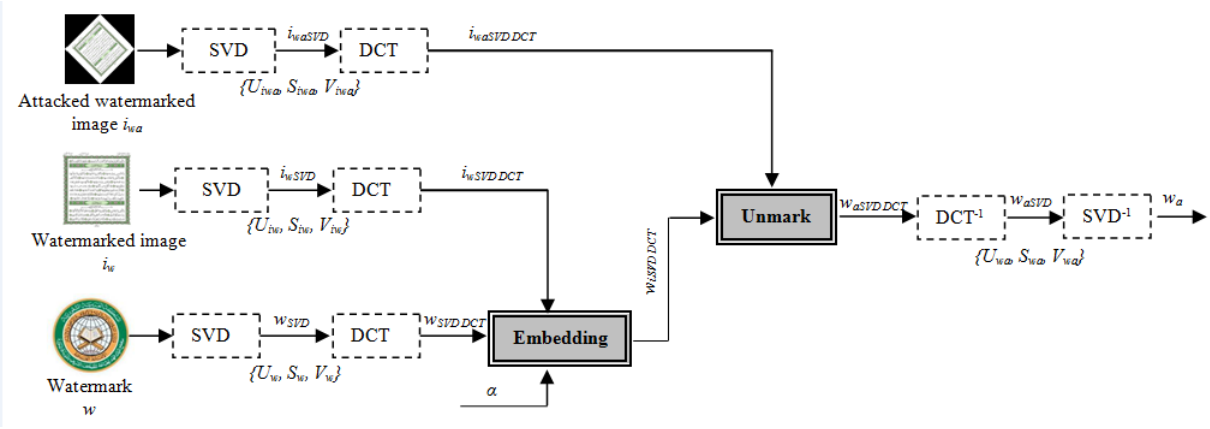


Figure 2: Watermark extraction algorithm

5 Tests and Robustness Evaluation

In this work, tests were conducted on a database of 120 text-image samples. The original images i [9] and the watermark w [9] are represented by colour text-images of size $256 \times 256 \times 3$. Figure 3 shows the images used in our tests by watermarking i with w using several values of the key α =0.1; 0.5; 0.98. We performed both the embedding/extraction processes only in the case of invisible watermarking, where $\alpha = 0.98$ (close to 1). The benchmark applied is that of the Stirmark benchmark from [8].

This benchmark is the most used attacks software and is well used by scientists in the digital-watermarking domain [8] for simulating attack scenarios.

We applied all the existing attacks from the Stirmark benchmark and obtained an attacked watermarked-image, i_{wa}, for each attack. We illustrate some of the most dangerous attacks, including: affine transformation, cropping, JPEG compression, median-filtering, additive-noise and rotation attacks, as shown in Table 1.

As known, attacks such as cropping, rotation and noise are considered as dangerous attacks due to their nature as a non linear transformation. The robustness of the proposed approach can be justified for two reasons: 1) The perfect invisibility of the watermark (in the embedding process); 2) embedding the watermark only in the singular values matrix S. the singular values matrix is known that contain the most important information in the image, which can very helpful to extract the watermark.

The robustness evaluation of the proposed approach is based on calculating the most known similarity measures. Hence, our performance evaluation cost-function consists of comparing the similarity degree PSNR and SSIM [7] between the original watermark and the extracted one following each attack. In Table 1, we have calculated those measures in the case of all the presented attacks. The PSNR measure is a critical metric in our experiments and is defined by the Equation (12):

$$PSNR = log_{10}(\frac{Max_w}{\sqrt{MSE}}) \tag{12}$$

Given that

$$MSE = \frac{1}{MN} \sum_{0}^{m-1} \sum_{0}^{n-1} \|w(i,j) - w_a(i,j)\|^2 \tag{13}$$

Where m, n are the image size and w(i,j), w_a(i,j) are the values of the pixels in the position (i,j). The SSIM calculation is given by Equation (13):

$$SSIM = \frac{(2\mu_w\mu_{wa} + c_1)(2\sigma_{wwa} + c_2)}{(\mu_w^2 + \mu_{wa}^2 + c_1)(\sigma_w^2 + \sigma_w^2 + \sigma_{wa}^2 + c_2)} \tag{14}$$

Given that:

μ_w and μ_{wa} are respectively the average of w and w_a.

$\sigma^2{}_w$ and $\sigma^2{}_w$ are respectively the variance of w and w_a.

σ_{wwa} is the covariance of w and w_a.

L is the dynamic range, of the pixel intensity (typically $2^{\#bitsperpixel}$ -1).

$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ with k_1=0.01 and k_2=0.03.

From Table 1, it is observed that the similarities between the original watermark and the extracted ones in the case of PSNR are in excess of 34db, which suggests a high similarity between the original watermark and the extracted watermark under those attack scenarios. For the SSIM, we see that its values are very close to 1, also suggesting that the original watermark and the extracted watermark are very similar. Moreover, the amplified differences in Table 1 demonstrate that visibly perfect watermark extraction was possible using our approach. We used the amplified difference to demonstrate where exactly the differences zones between the original watermark w and the extracted wa. Showing a simple difference between w and wa can not allow to distinguish visually the differences between w and w_a.

We compared the approach we propose regarding to some of the most remarkable works in the literature and

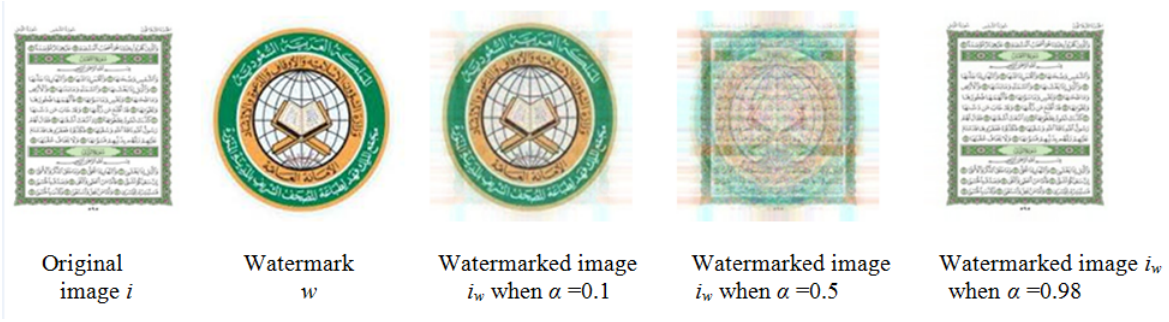
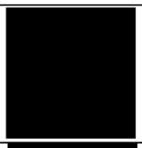
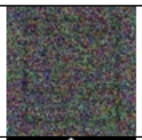


Figure 3: Watermark embedding with varying values of key (α)

Table 1: Results of extracted watermarks under several attack scenarios

Attack Type	Attacked watermarked image (i_{wa})	Extracted watermark (w_a)	Difference $\lvert w_a - w \rvert$	Amplified difference (+150)	PSNR (dB)	SSIM
Affine_7					45.21	0.971
Cropping_50					46.37	0.987
Jpeg_90					43.92	0.976
Median_9					46.11	0.984
Noise_80					36.72	0.954
Rotation_45					38.63	0.962

which are based on the combination of SVD and DCT watermarking. We note that the results obtained in our work in terms of PSNR measure is better, which means a higher robustness comparing to works bellow. The Table 2 shows a comparison of our approach with those works.

It should be noted that the proposed approach has a high complexity due to the quantity of information to embed presented by an entire image as a watermark. A compromise between robustness and hiding capacity is a major challenge in the literature. Unfortunately, till date there is no a watermarking scheme which can reach this compromise: Little information quantity to embed, low complexity and high robustness.

6 Conclusion

Copyright protection is a challenge that requires further research efforts in addition to the existing literature works. Multimedia protection is a topic of particular importance in recent years due to its economic and moral impacts. To address this problem, we propose a hybrid watermarking scheme based on SVD and DCT to ensure the originality and authenticity of text-images against illegal manipulations. It should be noted that many studies in this field have been presented but most of those studies present weaknesses at the robustness level, particularly against specific types of geometric attacks. The main contribution in this paper is found in the watermark extraction process that involves a third parameter; which is the attacked watermarked-image in the hybrid watermarking algorithm based on DCT and SVD. We tested our approach against many attack-types, and have only presented the most dangerous types in this paper (e.g. that include median-filtering, rotation, additive-noise attacks etc...). The most dangerous attacks consist to make alteration in the whole of the image (pixel by pixel) and not for specific zones.The robustness of our algorithm has been evaluated using widely known metrics from the literature: namely, the PSNR and SSIM metrics. The results obtained were very encouraging, allowing us to extract the watermark against attacks almost perfectly. The similarity between the original watermark and the extracted watermarks for each attack were very close. The only disadvantage of our approach algorithm was that it resulted with a higher complexity.

Acknowledgments

The authors would like to thank and acknowledge the IT Research Centre NOOR at Taibah University for their financial support during the academic year 2012/2013 under research grant reference number NRC1-126.

References

[1] M. Ali, C. W. Ahn, M. Pant, "A robust image watermarking technique using SVD and differential evolution in DCT domain", *Optik-International Journal for Light and Electron Optics,* vol. 125, no. 1, pp. 428–434, 2014.

[2] H. N. Andevari, S. Mirzakuchaki, "Image Watermarking Optimization in DCT-SVD Domain Using NSGA-II," *International Journal of Computer Theory and Engineering,* vol. 4, no. 2, pp. 309, 2012.

[3] G. Bhatnagar, B. Raman, "A new robust reference watermarking scheme based on DWT-SVD," *Computer Standards & Interfaces,* vol. 31, no. 5, pp. 1002–1013, 2009.

[4] S. W. Foo, Q. Dong, "A normalization-based robust image watermarking scheme using SVD and DCT," *World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering,* vol. 6, no. 1, pp. 205–210, 2010.

[5] F. Huang, Z. H. Guan, "A hybrid SVD-DCT watermarking method based on LPSNR," *Pattern Recognition Letters,* vol. 25, no. 15, pp. 1769–1775, 2004.

[6] Z. Li, K. H. Yap, B. Y. Lei, "A new blind robust image watermarking scheme in SVD-DCT composite domain," in *18th IEEE International Conference on Image Processing (ICIP),* pp. 2757–2760, 2011.

[7] L. Laouamer, O. Tayan, "An enhanced SVD technique for authentication and protection of text-images using a case study on digital Quran content with sensitivity constraints," *Life Science Journal,* vol. 10, no. 2, pp. 2591–2597, 2013.

[8] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, "Attacks on copyright marking systems," in *Proceedings of Second International Workshop on Information Hiding (IH'98)*, LNCS 1525, Springer-Verlag, pp. 219-239, 1998. (`http://www.petitcolas.net/fabien/watermarking/stirmark/`)

[9] Quran Complex, 2016. (`http://www.qurancomplex.org/`)

[10] D. Rosiyadi, S. J. Horng, P. Fan, et al. "Copyright protection for e-government document images," *IEEE Transactions on Multimedia,* vol. 19, no. 3, pp. 62–73, 2012.

[11] A. Sverdlov, S. Dexter, A. M. Eskicioglu, "Robust DCT-SVD domain image watermarking for copyright protection: embedding data in all frequencies," in *IEEE 13th European Signal Processing Conference,* pp. 1–4, 2005.

[12] X. Wu, W. Sun, "Robust copyright protection scheme for digital images using overlapping DCT and SVD," *Applied Soft Computing,* vol. 13, no. 2, pp. 1170–1182, 2013.

[13] X. Wu, W. Sun, "Robust copyright protection scheme for digital images using overlapping DCT and SVD," *Applied Soft Computing,* vol. 13, no. 2, pp. 1170–1182, 2013.

Table 2: Comparison of the proposed approach with some related works

Attack Scenario	PSNR in Proposed approach	PSNR in [1]	PSNR in [2]	PSNR in [6]
JPEG	43.92	31.92	48.3998	28.36
Rescaling	39.241	19.9781	37.1756	21.73
Gaussian Noise	36.72	14.2658	29.783	23.33
Median Filtering	46.11	13.9449	35.7747	25.52
Cropping	46.37	36.4068	12.2786	11.30
Rotation	38.63	6.4717	15.8719	12.32

[14] B. Wang, J. Ding, Q. Wen, et al. "An image watermarking algorithm based on DWT DCT and SVD," in *IEEE International Conference on Network Infrastructure and Digital Content,* pp. 1034–1038, 2009.

Lamri Laouamer is an assistant professor at the department of Management Information Systems, College of Business and Economics at Qassim University, KSA. He is also an associate researcher at the Laboratoire des Sciences et Techniques de Information, de la Communication et de la Connaissance (Lab-STICC), University de Bretagne Occidentale, Brest, France. He received his PhD in computer science, in the field of information security, from University de Bretagne Occidentale, France, in 2012; his MSc in computer science and applied mathematics from the University of Quebec at Trois Rivieres, Canada, in 2006; and his B.Sc. in computer science from the University of Setif, Algeria, in 1999. His research interests include multimedia watermarking, cryptology and information security. Dr. Lamri Laouamer is an associate editor of the Journal of Telecommunication Systems, published by Springer, and associate editor of the Journal of Innovation in Digital Ecosystems, published by Elsevier.

Omar Tayan completed his undergraduate degree in Computer and Electronic Systems from the University of Strathclyde, Glasgow, UK and his PhD in Computer Networks, Department of Electronic & Electrical Engineering from the same university. He currently works as an Associate Professor at the College of Computer Science and Engineering (CCSE) and IT Research Center for the Holy Quran and Its Sciences (NOOR) at Taibah University, Saudi Arabia. He was a consultant to the Strategic and Advanced Research and Technology Innovation Unit at the university and is one of the founding members of the "IT Research Center for the Holy Quran and Its Sciences (NOOR)" at Taibah University. His research interests include; Information Security, E-Learning technologies, performance modeling and simulation, high-speed computer networks and architectures, software simulation techniques and queuing theory, Wireless Sensor Networks for Intelligent Transportation Systems, Networks-on-Chip (NoC).

Anomalies Classification Approach for Network-based Intrusion Detection System

Qais Saif Qassim, Abdullah Mohd Zin, and Mohd Juzaiddin Ab Aziz
(Corresponding author: Qais Saif Qassim)

Research Center for Software Technology and Management, Information Science and Technology University Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia
(Email: qaisjanabi@gmail.com)
(Received Sep. 1, 2015; revised and accepted Dec. 7, 2015 & Jan. 15, 2016)

Abstract

Anomaly based intrusion detection system (A-IDS) is considered to be a better option than signature based system since it does not require prior knowledge of attack signature before it can be used to detect an intrusion. However managing alarms generated by this system is more difficult than signature-based intrusion detection systems (S-IDSs). This is due to the fact that S-IDS generates rich information along with the reported alarms whereas A-IDS may just identify the connection stream that is detected as malicious. A-IDS raises an alarm every time it detect an activity that deviates from the baseline model of the normal behaviour. Therefore, the cause of the anomaly itself is unknown to the intrusion detection system. This brings in a substantial challenge problem in managing IDS alarms and recognizing false positive from true alarms. Therefore, determining the class of an attack detected by anomaly-based detection systems is a significant task. This paper serves two folds; firstly, it presents a set of network traffic features that deemed to be the most relevant features in identifying wide range of network anomalies. Secondly, the paper presents an A-IDS alarm classifier based on machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system. Evaluation experiments showed that machine learning algorithms are capable of classifying malicious activities in an effective and efficient means.

Keywords: Alarm classification, anomaly-based, feature selection, machine learning

1 Introduction

Anomaly-based detection system is designed to uncover abnormal patterns of behaviors, in which anything that widely deviates from normal usage patterns will be considered as an intrusion [4]. It is considered to be a better option than signature based system since it does not require prior knowledge of attack signature before it can be used to detect an intrusion. However, identifying the class of attack poses a significant problem in anomaly based IDS. In signature based IDS, this process is trivial since each signature is a result of an analysis of the corresponding attack conducted by security experts; in which the attack class is manually assigned during the signature development process [9, 12]. Unlike signature-based IDS, the anomaly-based detection system cannot associate the detected activity with an attack class. In fact one of the major weaknesses of anomaly-based intrusion detection system is that, it cannot classify the detected activity to determine the severity level and the consequences of the detected activity [10].

By classifying an attack, it is possible to set default actions for handling a certain alarm. As well as, in order to estimate the risk of unknown attacks, a solution to automate the classification of anomaly-based alarms is required. However, so far no effective and efficient automatic or semi-automatic approach that is currently available, able to classify anomaly-based alarms at runtime [15, 31]. Thus, any anomaly-based alarm must be manually processed to identify its class; this may increase the workload of security analyst, and will effectively increase time required; as well as, the dependence on security analysts. Another limitation of manual alarm processing is that the complexity and dynamically changing traffic statistics may introduce the possibly of human error. This paper presents Network Anomalies Classifier (NAC) that uses machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system.

The rest of this paper is organized as follows: Section 2 presents an overview of the current state of attack identification and classification addressing the feature sets have been monitored, Section 3 presents the attack scenarios providing the common network traffic features to be monitored to identify different attack classes, Section 4 describes the research methodology, Section 5 discusses the evaluation of the proposed system and Section 6 presents

the conclusions and future works.

2 Related Works

IDS alarm classification has been an active research area for the past few years, recent researchers have focused on managing the generated alarms to identify real threats from false alarms and to classify the alarms into distinct classes. Several methods have been proposed to analyse the reported alarms based on different classification algorithms and network traffic features [1]. This section presents some of the recently proposed methods.

Entropy based analysis [21] have been employed to analyze a signature-based IDS alarms (more specifically, Snort) and detect real network attacks. The proposed method uses Shannon entropy to examine the distributions of five statistical features of Snort alarms as illustrated in Table 1. The features used are; the number of alarms generated from each distinct source IP address, the number of alarms sent to a destination host, source and destination threats' severity grade and datagram length. An adaptive false alarm filter [23] have been utilized to filter out false alarms with the best machine learning algorithm based on distinct network features. The Authors have intended to reduce the false alarms generated by signature-based IDS (Snort) in real time, and have selected 8 network features to represent the generated alarms as follows; Snort's description of an attack, attack classification, priority of an attack, packet type, source IP address, source port number, destination IP address and destination port number. They have used DARPA dataset to evaluate six different machine learning algorithms; K-nearest neighbor, support vector machines, naive bayes, neural networks and decision trees using Weka platform. And then, they have designed an adaptive false alarm filter to select the best single-performance algorithm in filtering out false alarms.

An approach of semi-supervised learning mechanism have been introduced by Chiu [3] to build an alarm filter for signature-based intrusion detection system. The authors have selected eight network features specifically; the connection's start time, the connection's duration, local and remote IP addresses that participated in the connection, connection's service, local and remote ports used by the connection, the number of bytes sent and received and the state of the connection. In [27] the author has used Lincoln laboratory dataset to find suitable subsets of features for network attack detection. The feature subsets were formed using prior knowledge from previous IDS researches and in addition, from analysing network attacks and their effect to the traffic flows, the selected features are illustrated in Table 1. The author showed that attacks of similar type, have similar effect to the network traffic and thus, subsets of features were formed for each attack type.

Flow-based analysis has been considered by Knuuti [16]. The author has compared the usability and performance of three different intrusion detection systems based on the identified network traffic flow features. The evaluated systems were Snort, Bro-IDS and TRCNetAD. Snort and Bro-IDS are signature-based intrusion detection systems while the later is an anomaly-based IDS. The features set that the author used are as illustrated in Table 1, which are statistical representations of the network traffic flow. The study conducted two, one week long, traffic capturing periods to collect data for the evaluation. Using the selected features, Snort was able to detect over 1.5 million intrusions during the one-week traffic capturing period. Snort was able to detect buffer overflow attacks, Trojan, denial of service, VoIP attacks, Heap overflow attacks, DNS spoofing attack and spyware. Bro-IDS detected approximately eight thousand intrusions which were address and port scan. TRCNetAD detected 150 thousand anomalies during the same time period.

Rule adaptation approach in managing IDS alarms have been considered by Lin [20]. The study has proposed a Weighted Score-based Rule Adaptation (WSRA) mechanism; which have the facility to learn from expert's feedback. Features used in this work are illustrated in Table 1 and as follows; total number of source and destination IP addresses in defined time window, source and destination port number, snort's signature, attack class, and timestamp.

Monitor deviations in network traffic features distributions from baseline model had been considered in IDS alarm management approaches [5]. The study analysed events that affect the distribution of traffic features and mark them as anomalies. The proposed system monitored network-wide backbone traffic using the features listed in Table 1. They have monitored the changes on the four IP packet header features between traffic flows using different algorithms. However, the study didn't evaluate the proposed method in real network traffic.

3 Feature Selection Based on Attack Scenarios

Feature selection is an important step in building intrusion detection and constructing alarm classification modules. During feature selection phase, a set of network traffic attributes or features deemed to be the most effective attributes is extracted in order to construct suitable classification module [29, 33]. A key challenging problem that many researchers face is how to choose the optimal set of features [1, 28], as not all features are relevant and have an impact on the classification performance, and in many cases, irrelevant features can impact the classification accuracy and cause slow training and testing processes. By analysing known attacks and their influence to the normal network traffic, it is possible to define which traffic features are relevant and therefore should be monitored. The idea behind this approach is to define the characteristics of a specific attack category. This is done by analysing

Table 1: Network traffic features used in prior studies

Study	Features Used	Num. of Features
[21]	The number of alarms generated from each distinct source IP address, the number of alarms sent to a destination host, source and destination threats' severity grade and datagram length	5
[23]	Description of the attack, Snort's classification, Alarm priority, packet type, source IP address, source port number, destination IP address and destination port number.	8
[3]	The connection's start time, the connection's duration, local and remote IP addresses that participated in the connection, connection's service, local and remote ports used by the connection, the number of bytes sent and received and the state of the connection	8
[27]	IP address, timestamp, number of receiving sequences, number of receiving sequences from different IP's, number of sending sequences, number of sending sequences to different IP's, amount of data received, amount of data sent, amount of packets received, amount of packets sent, number of different port numbers used over 1024, number of port numbers used over 1024, number of different port numbers used below or at 1024, number of port numbers used below or at 1024, number of UDP flows, number of TCP connections, number of ICMP packets, number of SMTP connections, number of FTP connections, number of HTTP connections, number of DNS connections, number of Telnet connections, number of SSH connections	24
[20]	Total number of source and destination IP addresses in defined time window, source port number, destination port number, snort's signature, attack class, and timestamp.	5
[16]	IP address, timestamp, number of ICMP packets, number of UDP flows, number of TCP connections, amount of received data, amount of sent data, number of received packets, number of sent packets, number of different port numbers used over 1024, number of port numbers used over 1024, number of different port numbers used below 1024, number of port numbers used below 1024, number of receiving sequences from different IP's, number of receiving sequences, number of sending sequences to different IP's and number of sending sequences.	17
[5]	Source IP address, destination IP address, source port number and destination port number.	4

the attacks classification done by MITRE Corp [24]. Researchers at MITRE Corp. have developed attack taxonomy for the United State Department of Homeland Security [7]; the main goal of this taxonomy is to create a list of patterns employed by attackers when compromising information systems, along with a comprehensive schema and classification taxonomy [34]. The project entitled as the Common Attack Pattern Enumeration and Classification (CAPEC). The classification in CAPEC is based on the mechanism used to attack that include; resource depletion, network reconnaissance, spoofing, exploitation of authentication, and exploitation of privileges.

3.1 Resource Depletion (DOS)

An attacker depletes a resource to the point that the target's functionality is affected. The result of a successful resource depletion attack is usually the denial of one or more services offered by the target [11, 19]. In order to deplete the target's resources the attacker must interact with the target and a client or script capable of making repeated requests over a network. If the attacker has some privileges on the system the required resource will likely be the ability to run a binary or upload a compiled exploit, or write and execute a script or program that consumes resources. Most of resource depletion attacks are detectable by monitoring from the traffic flows and the amount of data sent by the source. Therefore, the features that should be monitored for resource depletion

attacks are as follows [26, 27];

1) Number of sequences received during the observation period;
2) Amount of bytes received during the observation period;
3) Total number of packet received;
4) Total number of sequences received during the observation period from different IP's;
5) Number of sequences sent during the observation period;
6) Amount of bytes sent during the observation period;
7) Total number of packet sent;
8) Total number of sequences sent during the observation period to different IP's;
9) Total number of different TCP and UDP port numbers used by source;
10) Total number of different TCP and UDP port numbers used by the host;
11) Number of TCP requests for transmission;
12) Number of half open connections;
13) Number of established connections which represents an open connection;
14) Number of connection termination requests sent;
15) Number of confirming connection termination received;
16) Total number of TCP connections during the observation period;
17) Total number of UDP flows during the observation period;
18) Total number of TCP connections initiated by source;
19) Total number of UDP flows received;
20) Total number of TCP connections initiated by the host;
21) Total number of UDP flows sent.

3.2 Network Reconnaissance (Probe)

An attacker engages in network reconnaissance operations to gather information about a target network or its hosts. Network Reconnaissance techniques can range from stealthy to noisy and utilize different tools and methods depending upon the scope of the reconnaissance [24, 26]. Host discovery and port scanning are common examples of network reconnaissance, where the attacker tries to map out IP addresses and operating systems that are in use, as well as what services the hosts are providing [14]. In general, in network reconnaissance operations the attacker tries to find out all the possible means and methods that it can use to perform other attacks such as denial of service or gaining an unauthorised access to the inner network. Most of network reconnaissance attacks are detectable by monitoring from the traffic flows. Therefore, the features that should be monitored for such attacks are as follows [17, 23];

1) Number of sequences received during the observation period;
2) Total number of sequences received during the observation period from different IP's;
3) Number of sequences sent during the observation period;
4) Total number of sequences sent during the observation period to different IP's;
5) Total number of different TCP and UDP port numbers used by source;
6) Total number of different TCP and UDP port numbers used by the host;
7) Number of half open connections;
8) Number of connection termination requests sent;
9) Number of confirming connection termination received;
10) Total number of TCP connections during the observation period;
11) Total number of UDP flows during the observation period;
12) Total number of TCP connections initiated by source;
13) Total number of UDP flows received;
14) Total number of TCP connections initiated by the host;
15) Total number of UDP flows sent.

3.3 Spoofing

An attacker interacts with the target in such a way as to convince the target that it is interacting with some other principal and as such take actions based on the level of trust that exists between the target and the other principal [30]. Many of the protocols in the TCP/IP suite do not provide mechanisms for authenticating the source or destination of a message. They are thus vulnerable to spoofing attacks when extra precautions are not taken by applications to verify the identity of the sending or receiving host. IP spoofing may be used to leverage man-in-the-middle attacks against hosts on a computer network. Spoofing attacks which take advantage of TCP/IP suite protocols may be mitigated with deep packet inspection. The features that should be monitored for such attacks are as follows [25];

1) Total number of sequences received during the observation period from different IP's;
2) Total number of sequences sent during the observation period to different IP's;
3) Number of privileged port numbers used during the observation period;
4) Number of different privileged port numbers used during the observation period;
5) Number of registered ports used during the observation period;
6) Number of different registered port numbers used;
7) Total number of different TCP and UDP port numbers used by source;
8) Total number of different TCP and UDP port numbers used by the host;
9) Number of TCP requests for transmission;
10) Number of half open connections;
11) Number of established connections which represents an open connection;
12) Number of connection termination requests sent;
13) Number of confirming connection termination received.

3.4 Exploitation of Authentication

An attacker actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage identity and authentication. Such exploitation can lead to the complete subversion of any trust the target system may have in the identity of any entity with which it interacts. The exploitation of authentication attacks are detectable from the payload data by looking for specific patterns. Some of the attacks are though also detectable from the network traffic by looking for malformed packets that are oversized, fragmented or using, for example, abnormal TCP flag options [22]. Therefore, the features that should be monitored for such attacks are as follows;

1) Total number of sequences received during the observation period from different IP's;
2) Number of privileged port numbers used during the observation period;
3) Number of different privileged port numbers used during the observation period;
4) Number of registered ports used during the observation period;
5) Number of different registered port numbers used;
6) Number of half open connections;
7) Total number of TCP connections during the observation period;
8) Total number of UDP flows during the observation period;
9) Total number of TCP connections initiated by source;
10) Total number of UDP flows received;
11) Total number of TCP connections initiated by the host;
12) Total number of UDP flows sent.

3.5 Exploitation of Privilege/Trust

An attacker actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage access to its resources or authorize utilization of its functionality. Such exploitation can lead to the complete subversion of any control the target has over its data or functionality enabling almost any desired action on the part of the attacker. Similarly to exploitation of authentication attacks, this type of attacks detectable from the payload data by looking for specific patterns. However, some of the attacks are though also detectable from the network traffic. Therefore, the features that should be monitored for such attacks are as follows [35, 36];

1) Total number of sequences received during the observation period from different IP's;
2) Total number of sequences sent during the observation period to different IP's;
3) Number of privileged port numbers used during the observation period;

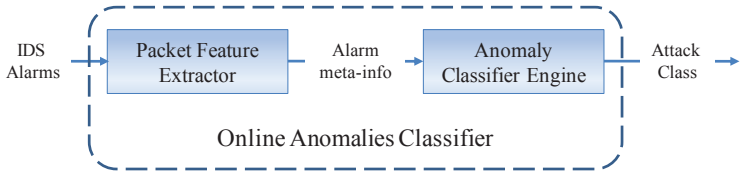


Figure 1: Online anomalies classifier

4) Number of different privileged port numbers used during the observation period;

5) Number of registered ports used during the observation period;

6) Number of different registered port numbers used;

7) Number of half open connections;

8) Total number of TCP connections during the observation period;

9) Total number of UDP flows during the observation period.

4 Network Anomalies Classifier (NAC)

This section presents an A-IDS alarm classification method which relies on machine learning algorithm and attack examples learnt from S-IDS during the training process. The proposed method monitors the network communication pattern and actively extracts the required network traffic features. The proposed system analyse IDS alarms and attempt to classify them based on prelearnt classification model. The classification model is constructed based on attack examples supplied during training phase, during the training phase Snort have been used to provide alarm class definitions of the activities detected by the anomaly detection system. The proposed system is represented by the Network Anomalies Classifier (NAC) module depicted in Figure 1. The NAC is responsible for an automatic classification of activities detected by a packet header-based anomaly detection system (specifically, PHAD) based on predefined set of patterns of attack mechanisms.

The proposed network anomalies classifier uses machine learning algorithm to assign class labels to the detected activities. The NAC consist of two interacting components; the Packet Features Extractor (PFE) and Anomaly Classifier Engine (ACE) as illustrated in Figure 1. The PFE monitors network traffic flow and extracts traffic flow features to generate alarm meta-information as a vector representing symptoms vector. The symptoms vector then, to be directed to the anomaly classifier engine for further analysis. The most suitable traffic flow features are selected by handpicking from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

The ACE is responsible for automatically classify the detected activity and determine the attack class. Before the classifier engine is able to classify new incoming alarms automatically, the ACE is trained with several types of attack meta-information to build a classification model. During the training phase, the attack meta-information is provided automatically by extracting specific information from known attack signatures. In this work, a signature-based IDS is deployed next to the anomaly detection system and both monitor the same network traffic. Consequently, the S-IDS is responsible to feed the NAC with the attack class of any alarm generated by the two systems.

4.1 Packet Features Extractor (PFE)

Network traffic contains features that are redundant or their contribution to the classification process is little. Therefore, it is essential to choose among the data what is relevant to consider and what is not [8]. By reducing the amount of features, the classifier's computational speed is improved and the overall performance is increased. Thus, Feature selection plays an important role when creating a model of the network traffic. The features should represent the traffic data as accurate as possible. The challenge is on discovering the most suitable features having major contribution to the classification process [4].

Network traffic is collected based on either packet data or network traffic flow, each provides a different type of visibility and collectively can provide a complete view of the network activities. As data streams flow across the network, the network packet-based sniffer captures each packet and decodes the packet's raw data, showing the values of various fields in the packet. The network traffic contains users' confidential information [33]. Consequently, a deep packet analysis cannot be done, and only limited analysis for the network traffic can be achieved. Therefore, the header fields of the packets can be checked, but not the user's data in the payload.

A traffic flow can be described as; all network packets with the same source/destination IP address, source/destination ports, protocol interface and class of service are grouped into a flow. Traffic flow is summarized data that provides a simple, effective, and scalable way to gain visibility into traffic types and bandwidth usage on the network. One important fact about network flows is that flows do not provide any packet payload. Rather, only meta-information about network connections is collected. The meta-information contains several attributes (e.g., the packets or bytes transferred in a flow). Unlike packet data approach, since network flows do not carry packet payload, all information which was transported in the original payload is irretrievably lost. While the lack of payload contributed to some advantages such as privacy and scalability [13].

Based on the available information from the literature it seems that an efficient attack classification can be done by using the network traffic flow information. Recent researches showed that network traffic flows could improve the accuracy of attack classification [13, 18]. Therefore, the network traffic flow method has been used in this work to monitor network behaviour. There are many advantages in using flow data instead of packet data. The major advantage comes from protecting the privacy and the confidentiality of the protected network as well as the reduced need of storage space for the data, since network flows requires a one tenth of the original packet-based data which is a huge difference. Network traffic flow provides abstract overview of the network state, performance and behaviour which are required to train the anomalies classifier engine.

Two approaches were used to select the relevant features from the network traffic. Initially, an analysis of what information the field of literature holds on this topic; then an evaluation of different attack scenarios and how they affect the network traffic behaviour have been prepared. The most suitable traffic flow features are selected from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

4.1.1 Packet Features Selection

After analysing the features from the attack scenarios point of view and what have been utilized in the literature, it seemed that the features used by [16] are very comparable to the features that should be monitored for each attack class. Therefore the features used by [16] were chosen as well as some other related features obtained from the attack scenarios analysis. The features to be monitored are listed in Table 2. The selected feature set containing statistical information that reflects the amount of change within each time interval.

As illustrated in Table 2, twenty five features have been selected to be monitored. The selected features will be represented as a vector of 25 elements, where each element represents its designated value. At this stage the extracted vectors will be defined as the symptoms vectors. To expound on the functionality of the packet features extractor, the functional model of the proposed system is shown in Figure 2.

4.2 Anomaly Classifier Engine (ACE)

The anomaly classifier engine is responsible for automatically classify the detected activity and determine the attack class, based on predefined set of patterns of known attack mechanisms that are defined in the CAPEC and CVE databases. The PFE monitors network traffic flow and extracts traffic flow features to generate alarm meta-information as a vector representing symptoms vector. The symptoms vector is then passed to the anomaly classifier engine that automatically determines the attack class. The development of ACE goes through two stages. First, the ACE is trained with several types of attack symptoms vectors. Then, when the training is completed, the ACE is ready to classify new incoming alarms automatically.

During training phase, a signature-based IDS is deployed next to the A-IDS such that the two systems monitor the exact network traffic as illustrated in Figure 3. Once the A-IDS generates an alarm the anomaly classifier engine learns the alarm class from the signature-based system. The strategy of alarm labelling process is as follow; if A-IDSs' reported activity did not trigger the S-IDS to generate an alarm it shall be considered as false alarm otherwise the classification engine will acknowledge S-IDS classification of the detected activity. Once the training phase is over, the proposed system enters the classification phase. During this phase, the packet header extractor actively extracts network traffic flow features of A-IDS reported activities and the anomaly classifier engine classifies the events based on the learnt classification model. The ACE includes the algorithm used to classify attacks; machine learning technologies have been used for classification process, to automatically and systematically classify attacks detected by an anomaly-based intrusion detection system. Machine learning can help to automate tasks and provide predictions where humans have difficulties to comprehend large amount of data. One major benefit of machine learning is the generalization ability, in which it has the ability of an algorithm to function accurately on new, unseen examples after having trained on a learning data set.

4.2.1 Machine Learning Algorithm Selection

The choice of which specific learning algorithm should be used is a critical step. The classifier's evaluation is most often based on classification accuracy (the percentage of correct classifications divided by the total number of events in the data set). There are various techniques available used to calculate a classifier's accuracy. One technique is to split the training set by using two-thirds for training and the other third for estimating perfor-

Table 2: Selected network traffic flow-based features (RD: Resource Depletion, NR: Network Reconnaissance, Spf: Spoofing, ExA: Exploitation of Authentication, ExP: Exploitation of Privilege/Trust)

Label	Feature	RD	NR	Spf	ExA	ExP
F1	Number of sequences received during the observation period	✓	✓			
F2	Amount of bytes received during the observation period	✓				
F3	Total number of packet received	✓				
F4	Total number of sequences received during the observation period from different IP's	✓	✓	✓	✓	✓
F5	Number of sequences sent during the observation period	✓	✓			
F6	Amount of bytes sent during the observation period	✓				
F7	Total number of packet sent	✓				
F8	Total number of sequences sent during the observation period to different IP's	✓	✓	✓		✓
F9	Number of privileged port numbers used during the observation period			✓	✓	✓
F10	Number of different privileged port numbers used during the observation period			✓	✓	✓
F11	Number of registered ports used during the observation period			✓	✓	✓
F12	Number of different registered port numbers used			✓	✓	✓
F13	Total number of different TCP and UDP port numbers used by source	✓	✓	✓		
F14	Total number of different TCP and UDP port numbers used by the host	✓	✓	✓		
F15	Number of TCP requests for transmission	✓		✓		
F16	Number of half open connections	✓	✓	✓	✓	✓
F17	Number of established connections which represents an open connection	✓		✓		
F18	Number of connection termination requests sent	✓	✓	✓		
F19	Number of confirming connection termination received	✓	✓	✓		
F20	Total number of TCP connections during the observation period	✓	✓		✓	✓
F21	Total number of UDP flows during the observation period	✓	✓		✓	✓
F22	Total number of TCP connections initiated by source	✓	✓		✓	
F23	Total number of UDP flows received	✓	✓		✓	
F24	Total number of TCP connections initiated by the host	✓	✓		✓	
F25	Total number of UDP flows sent	✓	✓		✓	

mance. In another technique, known as cross-validation, the training set is divided into mutually exclusive and equal-sized subsets and for each subset the classifier is trained on the union of all the other subsets. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier. If the error rate evaluation is unsatisfactory, the selected features must be re-examined.

Since the attack class and the related meta-information can be obtained, only supervised machine learning algo-

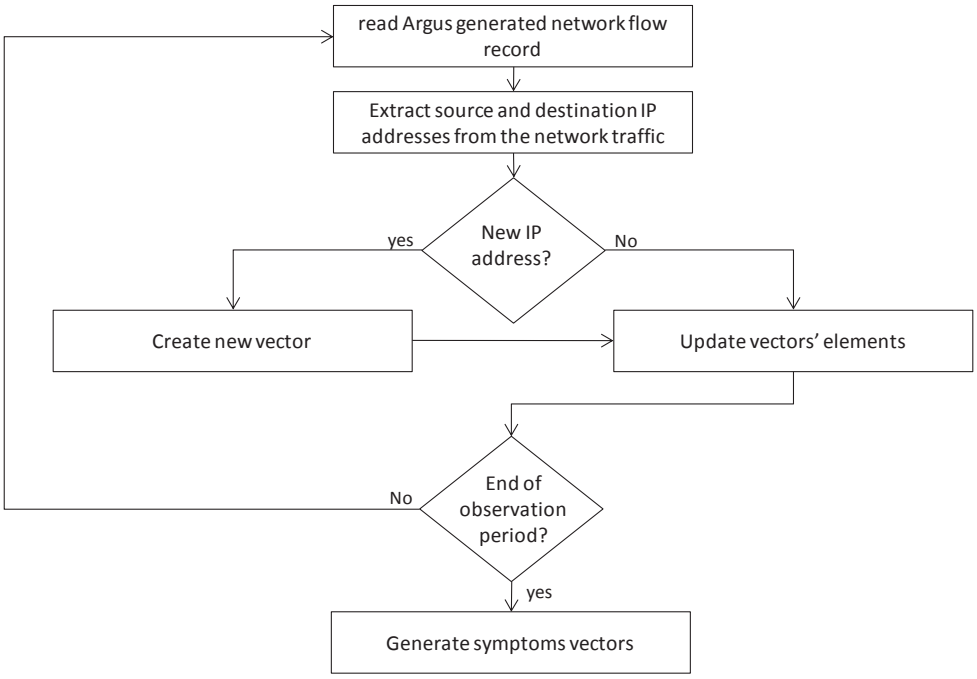


Figure 2: Functional model of the proposed method

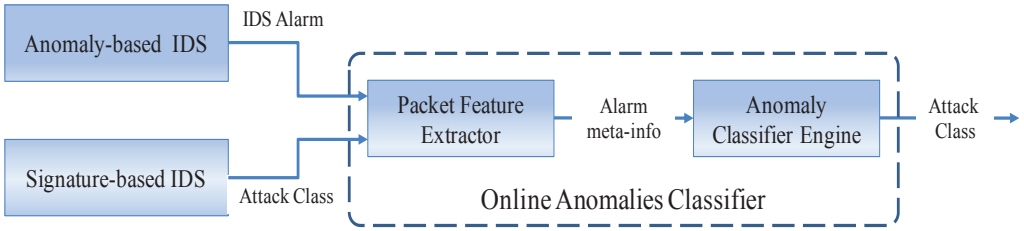


Figure 3: Online anomalies classifier during training phase

rithms have been considered in this work. These algorithms generally achieve better results than unsupervised methods. However, the classification algorithm must meet several requirements as listed in Table 3.

In this work, five machine learning algorithms have been considered as follows; Random Committee, Rotation Forest, PART, Random Forest and Random Tree. These algorithms implement supervised techniques, their training and classification phase are fast and able to handle large amount of data. In this work, machine learning evaluations have been implemented by using Weka platform. Weka is a well-known collection of machine learning algorithms, it also provide a comprehensive framework to execute benchmarks on several datasets under the same testing conditions.

Random Tree is a decision tree that considers number of randomly chosen attributes at each node. Random Tree have been introduced by [2] as a base classifier for his random forest classification algorithm. Random Tree develops un-pruned decision trees furthermore, it does not perform and optimization on its resultant rulesets.

Random Committee is an ensemble of randomized Random Tree classifiers. Each Random Tree classifier is built using a different random number seed. The final prediction is a straight average of the predictions generated by the individual base classifiers. Rotation Forest [32] have proposed an ensemble-classifier based on feature extraction. The model uses decision tree algorithms (J48) as base classifier and the feature extraction is based on Principal Component Analysis (PCA). PCA have been used to determine features feasibility and find out whether they do contribute to increased classification accuracy. In generating the training dataset, the feature set is randomly split into number of subsets and the Principal Component Analysis (PCA) is applied to each subset. The coefficients of the principal components is represented in a vector for each subset, and organized in a rotation matrix. All principal components are retained in order to preserve

Table 3: Machine learning selection criteria

Num.	Criteria	Description
1	Support for multiple classes	The attacks fall into five different categories. Therefore, it is required that, the selected algorithm supports multiclass classification.
2	Able to handle large amount of data	Using large amounts of memory can seriously degrade the system. Quite a few learning algorithms can be trained incrementally, one data row at a time. These methods generally have runtime that is linear in the number of rows and fields in the data and only require the current data row to be presented in the main memory. Because of this, they can process large amount of data.
3	High accuracy classification	One of the significant requirements is that, the machine learning algorithm should classify with high accuracy and low false positive and negative.
4	Able to train with small data set (fast training)	It is required that, the machine learning algorithm is able to develop the classification model in a small number of data set, to decrease the amount of alarms required.
5	Having an explicit underlying probability model	The machine learning algorithm should be based on statistical approaches, which provides a probability that an instance belongs in each class, rather than simply a classification.
6	Developed for academic researches	Because machine learning is beyond the scope of this work.

the variability information in the data. Thus, number of axis rotations takes place to form the new features for a base classifier. The proposed rotation forest ensemble have been evaluated on a selection of 33 benchmark data sets from the UCI repository and compared it with Bagging, AdaBoost, and Random Forest. The classification accuracy was more accurate than in AdaBoost and Random Forest, and more diverse than these in Bagging as well.

PART [6] have introduced PART rule-induction algorithm which utilized C4.5 and RIPPER rule-learning algorithms to propose a classification technique for inferring rules by repeatedly generating partial decision trees without the needs for complex optimization. It adapts the separate-and-conquer strategy in that it builds a rule, removes the instances it covers and continues creating rules recursively for the remaining instances until none are left. In essence to make a single rule, a decision tree is build for a selected set of instances, then the leaf with the largest coverage is made into a rule and that decision tree will be discarded. PART is a partial decision tree algorithm, which is the developed version of C4.5 and RIPPER algorithms. The main speciality of the PART algorithm is that it does not need to perform global optimisation like C4.5 and RIPPER to produce the appropriate rules; instead it utilises separate-and-conquer methodology to builds a partial C4.5 decision tree recursively and makes the "best" leaf into a rule.

Random Forests; is a combination of decision trees such that each constructed tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest. The concept behind the random forests is that, significant improvements in classification accuracy would achieve from growing an ensemble of trees furthermore each tree to vote for the most popular class. Random forests have been introduces by [2] and have been defined as an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

5 Evaluation Results of NAC

This section presents the evaluation results of the proposed network anomaly classifier. First, it describes the dataset employed and then the evaluation results are presented.

Table 4: Machine learning selection criteria

Attack Class	Dataset A	Dataset B	Dataset C	Dataset D	Dataset E
dos	615	1004	689	0	1378
u2r	15	0	1807	1084	3614
r2l	310	333	148	0	270
data	41	0	2	198	114
probe	171	13	144	148	195
Total	1152	1350	2790	1430	5571

5.1 Evaluation Dataset

The selected machine learning algorithms have been evaluated against five different datasets. The evaluation was based on the classification accuracy using the defined network traffic features. The datasets contain network traffic features representing network state during alarms identified by security analyst or raised by signature-based IDS (attack only dataset); each dataset contains number of instances representing network traffic audit records during a detected malicious activity as shown in Table 5 and Figure 4 illustrate the percentage distribution of attack types in datasets

Dataset A: This dataset contains 1152 instances, having majority of denial of service attacks by random selection. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 2:1, and the ratio of remote to local attacks and probe is also about 2:1. The dataset contains some attacks representing the user to root and data attacks. However, some classes have few audit records, which may impact negatively to the detection accuracy.

Dataset B: contains 1350 instances, having majority of dos attacks and some other attacks randomly selected, this dataset represents a scenario when an attacker uses probe and remote to user attacks to cause network resource unavailable to its intended users, which is common in real scenarios.

Dataset C: include 2790 instances represents a scenario when an attacker uses probe and remote to user attacks with dos to gain root privileges. Therefore, the dataset have a majority of user to root attacks. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 1:2.

Dataset D: include 1430 instances represents the same scenario of Dataset C when an attacker uses probe and remote to user attacks to gain root privileges but without the using of dos attacks.

Dataset E: This dataset contains 5571 instances randomly collected, having a majority of u2r attacks. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 2:1.

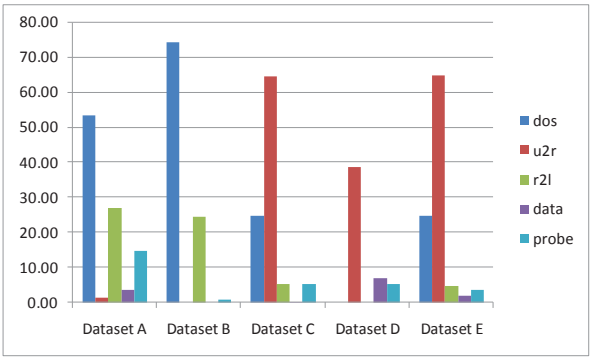


Figure 4: Percentage distributions of attack types in datasets

5.2 Evaluation Result

Three performance metrics have been used for machine learning comparison, classification accuracy, Precision and F-Measure. The performance of the selected machine learning algorithms have been conducted by training and testing with above five datasets to show its performance in different scenarios. However, there are four factors which influence the classification accuracy; the number of samples (alarms) processed during training phase, the frequency distribution of the alarms, the machine learning used and the network traffic features used. Table 4 illustrates the detection accuracy of the five datasets using different machine learning algorithms. Based on the above results of five datasets, it can conduct that Random Committee and Random Tree perform better than other algorithms and their detection accuracy almost identical, but the precision of Random Committee is higher than Random Tree. Therefore, in this work Random Committee will be used to classify the detected activities.

6 Conclusions and Future Works

In order to estimate the risk of unknown attacks, a solution to automate the classification of anomaly-based alarms is required. However, So far no effective and efficient automatic or semi-automatic approach is currently available able to classify anomaly-based alarms at run-

Table 5: Machine learning selection criteria

Machine Learning	Dataset A	Dataset B	Dataset C	Dataset D	Dataset E
Random Committee	96.78%	99.85%	98.49%	99.23%	98.20%
Rotation Forest	94.18%	99.03%	97.88%	98.04%	98.09%
PART	93.22%	99.18%	97.13%	98.04%	98.06%
Random Forest	96.61%	99.70%	98.45%	99.09%	98.18%
Random Tree	96.78%	99.85%	98.49%	99.23%	98.20%

time. Thus, any anomaly-based alarm must be manually processed to identify its class; this may increase the workload of security analyst, and will effectively increase time required; as well as, the dependence on security analysts.

This paper presents Network Anomalies Classifier (NAC) that uses machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system. The concept behind the proposed methodology is that, attacks those share some common network traffic flow behaviors are usually in the same class. Based on the available information from the literature it seems that an efficient attack classification can be done by using the network traffic flow information. Recent researches showed that network traffic flows could improve the accuracy of attack classification. Therefore, the network traffic flow method has been used in this work to monitor network behaviour. Thus by extracting traffic flow sequences triggered by certain attack, it is possible to compare those sequences to previously collected data using machine learning algorithm, then to infer the attack class from the matching sequences.

Two approaches were used to select the relevant features from the network traffic. Initially, an analysis of what information the field of literature holds on this topic; then an evaluation of different attack scenarios and how they affect the network traffic behaviour have been prepared. The most suitable traffic flow features are selected by handpicking from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

In this work, five machine learning algorithms have been considered as follows; Random Committee, Rotation Forest, PART, Random Forest and Random Tree. Evaluation experiments showed that machine learning algorithms are capable of classifying malicious activities in an effective and efficient means. However, a too low number of samples could generate an inaccurate classification. Therefore, as the number of training samples increases, accuracy increases. Based on the evaluation experiments results, it can conduct that Random Committee and Random Tree perform better than other algorithms and their detection accuracy almost identical, but the precision of Random Committee is higher than Random Tree. Therefore, as future works random committee algorithm will be used to classify the detected activities to estimate the security risk level.

References

[1] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] C. Y. Chiu, Y. J. Lee, C. C. Chang, W. Y. Luo, and H. C. Huang, "Semi-supervised learning for false alarm reduction," in *Advances in Data Mining Applications and Theoretical Aspects,* Springer Berlin Heidelberg, vol. 6171, pp. 595–605, 2010.

[4] G. Fernandes and P. Owezarski, "Automated classification of network traffic anomalies," *Security and Privacy in Communication Networks*, vol. 19, pp. 91–100, 2009.

[5] R. Fontugne, T. Hirotsu, and K. Fukuda, "An image processing approach to traffic anomaly detection," in *ACM Proceedings of the 4th Asian Conference on Internet Engineering (Aintec'08)*, pp. 17, 2008.

[6] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 144–151, 1998.

[7] V. N. L. Franqueira, Z. Bakalova, T. T. Tun, and M. Daneva, "Towards agile security risk management in RE and beyond," in *IEEE Workshop on Empirical Requirements Engineering*, pp. 33–36, 2011.

[8] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: A survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 271, 2013.

[9] A. A. Ghorbani, W. Lu, and M. Tavallaee, "Network intrusion detection and prevention," *Information Security*, vol. 47, pp. 27–54, 2010.

[10] M. Guimaraes and M. Murray, "Overview of intrusion detection and intrusion prevention," in *Proceedings of the ACM 5th Annual Conference on Information Security Curriculum Development,* pp. 44–46, 2008.

[11] X. Hongbin and X. Wenbo, "Research on method of network abnormal detection based on hurst parameter estimation," in *Proceedings International Confer-*

ence on Computer Science and Software Engineering, vol. 3, pp. 559–562, 2008.

[12] G. Javadzadeh and R. Azmi, "IDuFG: Introducing an intrusion detection using hybrid fuzzy genetic approach," *International Journal of Network Security Its Applications*, vol. 17, no. 6, pp. 754–770, 2015.

[13] J. H. Jun, D. Lee, C. W. Ahn, and S. H. Kim, "DDoS attack detection using flow entropy and packet sampling on huge networks," in *The Thirteenth International Conference on Networks*, pp. 185–190, Nice, France, 2014.

[14] Y. Kim, J. Y. Jo, and K. K. Suh3, "Baseline profile stability for network anomaly detection," *International Journal of Network Securit*, vol. 6, no. 1, pp. 60–66, 2008.

[15] J. M. Kizza, "Introduction to computer network vulnerabilities," in *Guide to Computer Network Security*, vol. 4, pp. 87–103, 2015.

[16] O. Knuuti, T. Seppälä, T. Alapaholuoma, J. Ylinen, P. Loula, P. Kurnpulainen, and K. Hätönen, "Constructing communication profiles by clustering selected network traffic attributes," in *5th International Conference on Internet Monitoring and Protection (ICIMP'10)*, pp. 105–109, 2010.

[17] P. G. Kumar and D. Devaraj, "Network intrusion detection using hybrid neural networks," in *2007 International Conference on Signal Processing Communications and Networking*, pp. 563–569, 2007.

[18] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 217, 2005. Press.

[19] F. Y. Leu and Z. Y. Li, "Detecting DoS and DDoS attacks by using an intrusion detection and remote prevention system," in *Fifth International Conference on Information Assurance and Security*, pp. 251–254, 2009.

[20] H. H. Lin, C. H. Mao, and H. M. Lee, "False alarm reduction by weighted score-based rule adaptation through expert feedback," in *IEEE 2nd International Conference on Computer Science and its Applications*, pp. 1–8, 2009.

[21] T. Liu, Z. Wang, H. Wang, and K. Lu, "An entropy-based method for attack detection in large scale network," *International Journal of Computer, Communications and Control*, vol. 7, no. 3, pp. 242–250, 2012.

[22] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proceedings of the ACM Symposium on Applied Computing* , pp. 346, 2003.

[23] Y. Meng and L. Kwok, "Adaptive false alarm filter using machine learning in intrusion detection," *Practical Applications of Intelligent Systems*, pp. 573–584, 2011.

[24] Mitre Corporation, "Common attack pattern enumeration and classification (CAPEC)," 2011.

[25] N. Mohd, S. Annapurna, and H. S. Bhadauria, "Taxonomy on security attacks on self configurable networks," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 44–52, 2015.

[26] T. L. Nielsen, J. Abildskov, P. M. Harper, I. Papaeconomou, and R. Gani, "The CAPEC Database," *Journal of Chemical & Engineering Data*, vol. 46, pp. 1041–1044, 2001.

[27] A. Niemelä, "Traffic Analysis for Intrusion Detection in Telecommunications Networks," *Master of Science Thesis, Tampere University of Technology,* 2011.

[28] I. V. Onut and A. A. Ghorbani, "A feature classification scheme for network intrusion detection," *International Journal of Network Security*, vol. 5, no. 1, pp. 1–15, 2007.

[29] S. Parsazad, E. Saboori, and A. Allahyar, "Fast feature reduction in intrusion detection datasets," in *MIPRO Proceedings of the IEEE 35th International Convention*, pp. 1023–1029, 2012.

[30] Q. Qian, T. Wang, and R. Zhan, "Relative network entropy based clustering algorithm for intrusion detection," *International Journal of Network Security*, vol. 15, no. 1, pp. 16–22, 2013.

[31] O. Rodas and M. A. To, "A study on network security monitoring for the hybrid classification-based intrusion prevention systems," *International Journal of Space-Based and Situated Computing*, vol. 5, no. 2, pp. 115, 2015.

[32] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1619–30, 2006.

[33] N. Sharma and S. Mukherjee, "A layered approach to enhance detection of novel attacks in IDS," *International Journal of Advances in Engineering Technology*, vol. 4, no. 2, pp. 444–455, 2012.

[34] H. Wang, M. Guo, L. Zhou, and J. Camargo, "Ranking attacks based on vulnerability analysis," in *2010 43rd IEEE Hawaii International Conference on System Sciences*, pp. 1–10, 2010.

[35] K. Wang, S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, LNCS 3224, pp. 203–222, Springer, 2004.

[36] J. Yu and Y.V.R. Reddy, "TRINETR: an intrusion detection alert management systems," in *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 235–240, 2004.

Qais Saif Qassim is a Ph.D. candidate in Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). His research interest in network security and management.

Abdullah Mohd Zin received his PhD from the University of Nottingham, United Kingdom in 1993. He is currently the dean of Faculty of Information Science and Technology, UKM. His specialization in system

architecture, programming language, communication and distributed, formal method.

Mohd Juzaiddin Ab Aziz received his PhD degrees in Computer Science from University Putra Malaysia (UPM). Currently, he is the deputy dean of undergraduate studies in faculty science and information technology, UKM. His specialization in programming language technology and natural language processing.

An Improved Ownership Transfer and Mutual Authentication for Lightweight RFID Protocols

Peng-yu Cui
(Corresponding author: Peng-yu Cui)
Information Centre, Liaoning Geology Engineering Vocational College, Dandong 118008, China
(Email: cuipengyu1982@126.com)
(Received Sep. 24, 2015; revised and accepted Dec. 07, 2015 & Jan. 3, 2016)

Abstract

Radio Frequency Identification (RFID) technology is an automated identification technology which is widely used to identify and track all kind of objects. However, it is a challenging task to design an authentication protocol because of the limited resource of Lightweight RFID tags. Recently, a lightweight RFID authentication protocol and an ownership transfer of RFID tags are presented by Kulseng et al. Both protocols use Physically Unclonable Functions (PUF) and Linear Feedback Shift Registers (LFSR) which are well known lightweight operations. The number of gates which the protocols require can be significantly decreased and the most efficient protocol can be obtained with respect to the existing protocols. Unfortunately, their protocols face several serious security issues. In this paper, based PUF and LFSR, we suggest an improved mutual authentication and an improved ownership transfer for low-cost RFID Protocols. Security analysis shows that our protocol owns security and privacy.

Keywords: LFSR, mutual authentication, ownership transfer, PUF, RFID

1 Introduction

RFID (Radio Frequency Identification) is an emerging ubiquitous technology which identifies different kinds of objects based on radio wave signals. It has been widely used in many fields, such as inventory control, transportation payment, supply chain management and so on [11].

As many technologies, RFID faces also similar security concerns: Authentication, Confidentiality and Availability. For insecure RFID system, the user's privacy will face a great threat. An adversary can obtain user's privacy by eavesdropping or trace the tag's holder in such condition [15]. However, as RFID tags are generally low-cost device without tamper resistance, compromising RFID tag can be very easy. The challenge on addressing the security concerns is much harder than conversational technology [5].

Some authentication protocols have been suggested to use in RFID system which aiming to solve the privacy and forgery problems. Generally, we only consider the information security issues in the channel between tags and reader for research convenience because of the special property of tags. In order to promote the great potential of RFID technology, the cost of RFID tags must be competitive with existing solutions such as bar codes, which are very low-cost. Passive RFID tags with no battery have between 200-2000 hardware gates available for security measures. Unfortunately, traditional security mechanisms used in RFID system require a large number of gates. A low-cost version of AES has been shown to require 3,400 gates, while hash functions such as MD5 and SHA-256 have been implemented using between 8,000-10,000 gates. Therefore, it is a key problem for RFID system to design efficient and secure authentication protocol [1, 2, 14].

Many RFID authentication protocols based on Pseudo-Random Number Generator (PRNG operation) have been proposed to achieve security and privacy protection [3, 13]. Also, several light-weight RFID authentication protocols with inexpensive cryptographic primitives, such as XOR and hash functions, are also presented [8, 9, 10, 12]. However, these protocols suffer from either privacy and security issues or efficiency.

Physically Unclonable Functions (PUFs) are known as random functions that map challenges to responses. PUFs are unclonable because it computes random numbers with the help of the inherent variability of wire delays and gate delays in manufactured circuits [4]. The existence of the fact is that no two circuits have exactly the same delay properties, even if they were produced on the same wafer. Given a certain input, the tag's PUF will produce a certain output, while other tag's PUFs will produce different output.

Kulseng et al. [7] present a lightweight mutual authentication and ownership transfer protocol which can be considered as lightweight because their protocols do not require expensive cryptographic operations. Their protocols are basically designed by using Physically Unclonable

Functions (PUFs) and Linear Feedback Shift Registers (LFSRs) which are well known lightweight operations and are particularly suitable for the low-cost RFID tags. Their protocol requires only 784 gates for 64-bit variables. So, this protocol can certainly be considered to have a significant improvement. But, Kardas et al. [6] show that there are in fact several serious security issues with Kulseng et al.'s protocols.

In this paper, using PUFs and LFSRs, we give an improved mutual authentication and ownership transfer for lightweight RFID Systems. The remainder of this paper is organized as follows. In Section 2, we describe the lightweight mutual authentication proposed by Kulseng et al. and its drawbacks. Section 3 presents an improved mutual authentication and discusses its security. Section 4 proposes a new ownership transfer protocol. Concluding remarks are presented in Section 5.

2 Kulseng et al.'S Protocol and Its Drawbacks

The notations and steps for the protocol are described as follows.

2.1 Notations

- ID: Tag's ID which is unique.
- IDS: An index to tag's ID and is updated in each round.
- G_n: A greeting number.
- F: A random permutation function mapping within range [1, q], where log q is the bit length of IDS (LFSR can be used as F).
- P: A random permutation function mapping within a range [1, q] (P is implemented based on Physically Unclonable Functions (PUF)).

2.2 Description of Protocols

The initial ID, IDS and a random greeting number G_n are generated for each tag firstly. Then, G_{n+1} is computed by the PUF function stored in the tag as $G_{n+1} = P(G_n)$. The entry of (IDS, ID, G_n, G_{n+1}) are inserted into the backend database. The IDS, ID and G_n are stored in the tag. Kulseng et al.'s protocol consists of five steps as Figure 1.

1) The reader continuously broadcasts *Req* message.
2) Receiving *Req* from the reader, the tag responds with its IDS.
3) The reader looks up the corresponding greeting G_n for this tag. If it finds an entry, it computes $ID \oplus G_n$ and sends it to the tag.

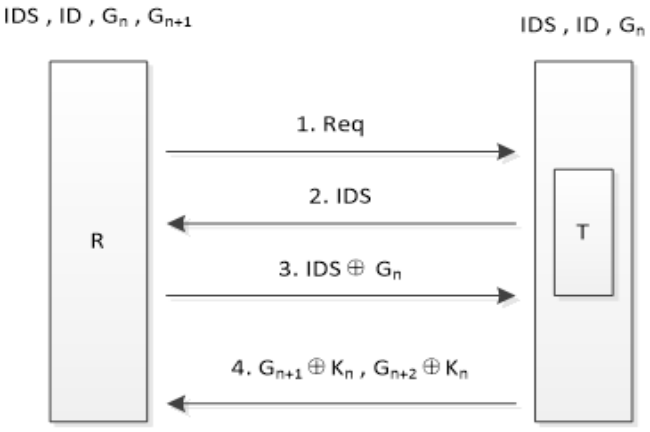


Figure 1: Kulseng et al.'s authentication protocol

4) Receiving the message $ID \oplus G_n$, the tag verifies the correctness of this response. If it is valid, it computes $G_{n+1} = P(G_n)$, $G_{n+2} = P(G_{n+1})$, $K_n = F(G_n)$ and $K_n^{'} = F(K_n)$. Then, it calculates $K_n \oplus G_{n+1}$, $K_n^{'} \oplus G_{n+2}$ and sends them to the reader. Finally, the tag updates $IDS = F(IDS \oplus G_n)$ and $G_n = G_{n+1}$.
5) The reader verifies $K_n \oplus G_{n+1} = F(G_n) \oplus G_{n+1}$. If it is valid, the reader can get G_{n+2} by $K_n^{'} \oplus G_{n+2} \oplus F(F(G_n))$. At last, it updates $IDS_{new} = F(IDS_{old} \oplus G_n)$, $G_n = G_{n+1}$ and $G_{n+1} = G_{n+2}$.

2.3 Security Analysis

Kardas et al. [6] describe three different security flaws of the authentication protocol above. Here, we introduce them briefly.

Set R as a legitimate reader, T as a legitimate tag and A as an adversary.

Message blocking attack.

1) R broadcasts Req and T sends its IDS to R.
2) R computes $ID \oplus G_n$ and sends it to T. Here, blocking attack occurs and transaction between R and T drops.
3) Then, A broadcast Req and T sends IDS to A.
4) A sends $ID \oplus G_n$ to T.
5) T calculates $K_n \oplus G_{n+1}$, $K_n^{'} \oplus G_{n+2}$ and sends them to A. After that, T updates $IDS = F(IDS \oplus G_n)$ and $G_n = G_{n+1}$.
6) T can no longer authenticate R because R will send $ID \oplus G_n$ and T has G_{n+1}. T will not verify ID.

Desynchronization attack.

The protocol can not assure integrity. When A inserts a random message to the second message at Step 4, the synchronization between R and T will be broken.

1) R broadcasts Req and T sends its IDS to R.
2) R computes $ID \oplus G_n$ and sends it to T.
3) T calculates $K_n \oplus G_{n+1}$, $K_n^{'} \oplus G_{n+2}$ and sends them to R. Here, A inserts random number n_x to the message $K_n^{'} \oplus G_{n+2}$, $K_n{'} \oplus G_{n+2} \oplus n_x$. Finally, T updates $IDS = F(IDS \oplus G_n)$ and $G_n = G_{n+1}$.
4) Receiving message $K_n \oplus G_{n+1}$ and the modified message $K_n^{'} \oplus G_{n+2} \oplus n_x$, R verifies whether $K_n \oplus G_{n+1}$ is valid. Because the message is correct, R updates $IDS = F(IDS \oplus G_n)$ and $G_{n+1} = K_n^{'} \oplus G_{n+2} \oplus n_x \oplus K_n^{'} = G_{n+2} \oplus n_x$.
5) In the next section, the R has G_n, G_{n+1} and $G_{n+1}^{'} \neq P(G_n)$. According to the protocol, T can authenticate R but R will not authenticate T.

The misuse of LFSR G.

Kardas et al. point that an adversary can easily find out the secret ID and trace the tag because of the use of LFSR. This attack can be accomplished as follows. Assume that an adversary observes a whole authentication session of a tag. The adversary who has listened the communication between the reader and the tag can obtain the following session messages: Req,IDS_{old},$ID \oplus G_n$,$G_{n+1} \oplus K_n$,$G_{n+2} \oplus K_n^{'}$.

Then, the adversary sends a fake query to the tag. The tag will response $IDS_{new} = F(IDS_{old} \oplus G_n)$ to the adversary. It is critical that $IDS_{old} \oplus G_n$ can be gotten from the value of $F(IDS_{old} \oplus G_N)$ easily. So, the adversary can deduce ID of the tag from IDS_{old}, $ID \oplus G_N$, $IDS_{old} \oplus G_n$ and trace the tag.

3 An Improved Mutual Authentication

In this section, an improved mutual authentication protocol is proposed as Figure 2 and its security is analyzed.

3.1 Notations

The notations are same as Section 2 and contain K_n. K_n is the share key between Reader and Tag.

3.2 An Improved Mutual Authentication Protocol

1) The reader continuously broadcasts Req message.
2) Receiving Req from the reader, the tag responds with its IDS and the tag updates $IDS = F(IDS \oplus G_n \oplus K_n)$.
3) According to IDS, if the reader finds an entry, it updates $IDS = F(IDS \oplus G_n \oplus K_n)$ firstly. Next step it generates a random number r and computes $K_n \oplus r$, $ID \oplus F(IDS \oplus r)$. Finally the reader sends $K_n \oplus r$, $ID \oplus F(IDS \oplus r)$ to the tag.

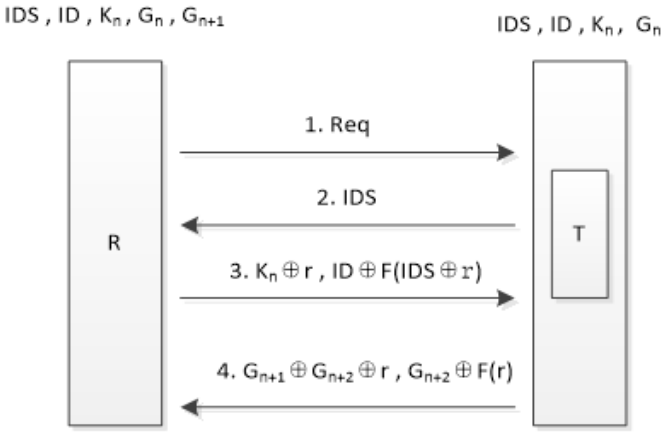


Figure 2: Our authentication protocol

4) Receiving the message $K_n \oplus r$ and $ID \oplus F(IDS \oplus r)$, the tag gets r from $K_n \oplus r \oplus K_n$ firstly. Next, it computes $F(IDS \oplus r)$. Furthermore the tag verifies the correctness of the ID from $ID \oplus F(IDS \oplus r) \oplus F(IDS \oplus r)$. If it is correct, the tag computes $G_{n+1} = P(G_n)$, $G_{n+2} = P(G_{n+1})$. Next step, it sends $G_{n+1} \oplus G_{n+2} \oplus r$ and $G_{n+2} \oplus F(r)$ to the reader. Finally, the tag updates $K_n = K_n \oplus F(K_n \oplus r)$ and $G_n = G_{n+1}$.
5) According to the message of the fourth step, the reader gets G_{n+2} from $G_{n+1} \oplus G_{n+2} \oplus r \oplus G_{n+1} \oplus r$ firstly. Next the reader verifies $G_{n+2} = G_{n+2} \oplus F(r) \oplus F(r)$. If it is correct, the reader updates $K_n = K_n \oplus F(K_n \oplus r)$, $G_n = G_{n+1}$, $G_{n+1} = G_{n+2}$.

3.3 Security Analysis

In this part, we present the security analysis of our scheme. In addition to limited storage capacity, low computational and communicational cost, our protocol withstand against modification attack, de-synchronization attack, disclosure attack, replay attack, man in middle attack, backward security, forward security, cloning attack and also achieve mutual authentication, tag anonymity and indistinguishability.

1) Resistance to modification attack.
No matter what parts of the messages in our protocol are modified, the reader or the tag can find it because IDS,G_n,G_{n+1},K_n is dynamic and IDS is random for adversary. So the reader or the tag can confirm each message in our protocol is modified and the protocol will be halted and the attacker can not get any valuable information.
2) Resistance to de-synchronization attack.
An attacker may try to desynchronize IDS,G_n between reader and tag. For this purpose, he blocks

messages from tag to reader in the fourth pass of the protocol. In order to handle this synchronization issue, it is suggested that the previous IDS,G_n value are stored in the tag side. When the IDS is not stored to the database, the reader will ask the tag to use old IDS,G_n.

3) Resistance to disclosure attack.
The key idea of the disclosure attack is that an attacker can slightly modify the challenge from the reader and then infer partial information from the response of the tag. In our protocol, the reader and tag have confidential data contain ID,G_n,G_{n+1},K_n all transmitted messages are random and secrecy. So attacker slightly modifies any challenge in all message, this protocol will be halted and he cannot get any useful information finally. As a result, this attack does not work on our protocol.

4) Resistance to replay attack and man in middle attack.
An attacker may try to do a replay attack by eavesdropping legitimate interactions. If an attacker wants to disguise reader, he replays first and third message. However, he cannot succeed because IDS will be both updated each round and random, and the third message is relation with IDS. So the tag can find this attack quickly. If an attacker wants to disguise tag, he replays second and fourth message. Also, he will not succeed because ID,G_n,G_{n+1},K_n have been updated and the message is relation with these parameters. The reader will not authenticate the disguised tag for the adversary replayed message is outdated. And, when the adversary tries the man-in-the middle attack, he will not succeed because the second message, the third message and the fourth message are dynamic, and lack of necessary parameters such as ID,G_n,G_{n+1},K_n.

5) Backward security and forward security.
It is essential that the previously transmitted information cannot be traced using the present transmission tag information, and the future information cannot be confirmed using the present transmission tag information. If the past and future location of the specific tag owner can be traced using the present information, it constitutes a serious privacy infringement. The proposed protocol prevents an adversary from acquiring tag information, by providing confidentiality based on unpredictable variations in the response message of the tag by every session. Moreover, IDS is updated each time and random for an adversary and G_n,G_{n+1},K_n is updated when authentication of the reader is complete and the tag is closed successfully, and the value of r is determined randomly by the reader, thus it guarantees backward security and forward security by disconnecting the relation with both the previously transmitted information and the future information.

6) Cloning attack.
To prevent cloning attacks, our protocol uses a unique PUF in tag. It is infeasible to construct two PUFs with the same challenge-response behavior. So an adversary can copy the PUF and cloning attack is invalid in our protocol.

7) Anonymity and indistinguishability.
Anonymity means that the attacker cannot identify the identity of tag and cannot track tag. Indistinguishability means that information emitted by tag should not be discriminated from other tags. The proposed protocol protects the information necessary for tag authentication by using the PUFs, LFSR function and the Random Number Generator, and guarantees that only the authenticated object knowing ID,G_n,G_{n+1},K_n can verify the information. Furthermore, as mentioned earlier, the proposed protocol is secure against backward security and forward security, and guarantees anonymity and indistinguishability.

8) Mutual authentication.
The proposed protocol provides mutual authentication between reader and tag. The tag authenticates reader by the value of ID and the reader authenticates tag by the value of G_{n+1}. The proposed protocol satisfies all the security requirements, and completely solves the privacy and forgery problems of the RFID system.

Next, we discuss this protocol about attacks described in Section 2.

Message blocking attack.
The way to resist blocking attack is same as that method in resistance to desynchronization attack. IDS,G_n are asked to store in the tag. If the database does not look up IDS, the reader can ask the tag to use old IDS,G_n to continue this protocol.

Desynchronization attack.
If an attacker attempts to insert any message to desynchronize this protocol, it can not gain its ends as all the elements in the message are linked together and any part changed will be found by reader or tag. For example, a random number n_x is inserted to the message in Step 4 as $G_{n+1} \oplus G_{n+2} \oplus r$, $G_{n+2} \oplus F(r) \oplus n_x$. According to our protocol, this attack has no effect because the reader will find $G_{n+2} \neq G_{n+2} \oplus F(r) \oplus n_x \oplus F(r)$.

The misuse of LFSR.
In our protocol, the use of LFSR will not leak any useful information of the tag. If an attacker obtains all messages in a whole authentication session between reader and tag. Then, the attacker sends a fake query to the tag. The tag wills response $IDS_{news} = F(IDS_{old} \oplus G_n \oplus K_n)$ to the adversary. He can obtain $IDS_{old} \oplus G_n \oplus K_n$ from

$F(IDS_{old} \oplus G_n \oplus K_n)$ and further to get $G_n \oplus K_n$. However, it is of no value. It is hard trace the tag only by both $G_n \oplus K_n$ and the previously transmitted information. 1summarizes the comparison of our protocol with Kulseng et al.'s protocol. Y is owns the ability of resistance to attack, N is not owns.

Table 1: Comparison of our protocol with Kulseng et al.'s protocol

Protocol	Our protocol	Kulseng et al.'s protocol
Modification attack	Y	N
De-synchronization attack	Y	N
Disclosure attack	Y	N
Replay attack and man in middle attack	Y	N
Backward security and forward security	Y	N
Cloning attack	Y	Y
Anonymity and indistinguishability	Y	N
Mutual authentication	Y	N

4 Ownership Transfer Protocol

In this section, we introduce Kulseng et al.'s ownership transfer protocol and attacks on it firstly. Then we present improved ownership transfer protocol.

Kulseng et al. [7] proposed two ownership transfer protocols. The first protocol assumes the existence of a trusted authority by both the reader and the tags, named the Trusted Third Party (TTP). The second ownership transfer protocol involves no third party. The authenticated reader that accesses the tag is called as owner. An ownership transfer protocol should satisfy the following two properties:

1) The old owner should not be able to access the tag after the ownership transfer takes place.

2) The new owner should be able to perform mutual authentication with the tag after the ownership transfer has taken place.

4.1 Kulseng et al.'s Ownership Transfer Protocol with TTP

The communications between the TTP and the readers are assumed to be secure. The old owner first gives its stored tuple (IDS, ID, G_{n+1}) to the new owner. It also transfers the verification pair G_n, G_{n+1} to the TTP. A secret value of PIN is securely shared between the TTP and the tag. The PIN is preinstalled in the tag hardware during production and is not accessible to anyone.

1) The new reader sends G_{n+1} to the TTP via a secure channel.

2) The TTP verifies whether the received G_{n+1} from the new reader equals to the one received from the previous owner, if so, then the new reader gets authenticated. Then the TTP sends $K_n \oplus G_n \oplus PIN$ to the reader, where $K_n = F(PIN)$.

3) The reader forwards the messages to the tag.

4) The tag computes $K_n = F(PIN)$ and gets G_n from $K_n \oplus G_n \oplus PIN$. If the computed G_n equals that it stores, the tag computes $G_n^{'} = P(G_{n+2})$,$G_{n+1}^{'} = P(G_n^{'})$ and $K_n^{'} = F(K_n)$,$K_n^{''} = F(K_n^{'})$. At last, tag calculates $K_n^{'} \oplus G_n{'}$,$K_n^{''} \oplus G_{n+1}^{'}$ and $K_t = F(G_n \oplus G_{n+1})$ and sends them to the reader.

5) The reader forwards these messages to the TTP.

6) Upon receiving the messages, the TTP verifies the correctness of the value K_t. Then it computes the random numbers $K_n{'}$,$K_n{''}$, and obtains the values of the pair value of $G_n{'}$,$G_{n+1}^{'}$ and sends them back the new owner via the secure channel. Now the new reader can start a new mutual authentication with the tag.

7) Both the TTP and the tag can update the PIN internally as $PIN_{new} = F(PIN_{old} \oplus G_n)$.

4.2 Attacks on Protocol

Now we show that the protocol above does not satisfy two secure properties.

- The old owner can access the tag after the ownership transfer takes place.
  Kardas et al. point that privacy of the tag can be elaborated by the old owner [6]. The old owner still knows ID of the tag because ID is constant and unique for each tag. Assume that the old owner A has recorded an successful session between R and T and a subsequent query to the tag T.

  1) A records all messages exchanged between R and T.
  2) A get G_n by $G_n = G_n \oplus ID \oplus ID$(the third message XOR ID).
  3) Next, A derives G_{n+1} by computing $G_{n+1} = (G_{n+1} \oplus K_n) \oplus F(G_n)$.
  4) A sends a fake query to the tag T and T sends back the updated IDS value.
  5) A computes $F(IDS \oplus G_n))$ by using G_n and IDS. Then, A verifies whether this value is equal to IDS which is received from the query. If they are equal, this session belongs to the T.

- The new owner can not perform mutual authentication with the tag after the ownership transfer has taken place.
  We introduce an attack that can make the new owner not implement mutual authentication with the tag. A malicious adversary A injects random numbers n_x and n_y to the message as $K_n^{'} \oplus G_n^{'} \oplus n_x$, $K_n^{''} \oplus G_{n+1}^{'} \oplus n_y$ and $K_t = F(G_n \oplus G_{n+1})$. According to this protocol, what the new reader holds are $G_n^{'} \oplus n_x$, $G_{n+1}^{'} \oplus n_y$, not really $G_n^{'}$, $G_{n+1}^{'}$. So, the new owner is not able to perform mutual authentication with the tag.

4.3 Improved Ownership Transfer Protocol with TTP

Here, we present an improved ownership transfer protocol with TTP.

1) The new reader sends G_{n+1} to the TTP via a secure channel.
2) The TTP verifies whether the received G_{n+1} from the new reader equals to the one received from the previous owner, if so, then the new reader gets authenticated. Then the TTP sends $PIN^{'} \oplus G_n \oplus PIN$ to the reader, where $PIN^{'} = F(PIN)$.
3) The reader forwards the messages to the tag.
4) The tag computes $PIN^{'} = F(PIN)$ and gets G_n from $K_n \oplus G_n \oplus PIN$. If the computed G_n equals that it stores, the tag computes $G_n^{'} = P(G_{n+2})$, $G_{n+1}^{'} = P(G_n^{'})$ and $K_n^{'} = F(G_n^{'})$, $K_{n+1}^{'} = F(G_{n+1}^{'})$. At last, tag calculates $PIN \oplus G_n^{'}$, $K_n{'} \oplus G_n^{'}$, $PIN \oplus G_{n+1}^{'}$, $K_{n+1}^{'} \oplus G_{n+1}^{'}$ and $K_t = F(G_n \oplus G_{n+1})$ and sends them to the reader.
5) The reader forwards these messages to the TTP.
6) Upon receiving the messages, the TTP verifies the correctness of the value K_t. Then it derives $G_n^{'}$, $G_{n+1}^{'}$ from $PIN \oplus G_n^{'}$, $PIN \oplus G_{n+1}^{'}$. Finally, it verifies $F(G_n^{'}) \oplus G_n^{'} \stackrel{?}{=} K_n^{'} \oplus G_n^{'}$ and $F(G_{n+1}^{'}) \oplus G_{n+1}^{'} \stackrel{?}{=} K_{n+1}^{'} \oplus G_{n+1}^{'}$. If both of them are correct, the TTP and sends $G_n^{'}$ and $G_{n+1}^{'}$ to the new owner via the secure channel. Now the new reader can start a new mutual authentication with the tag.
7) Both the TTP and the tag can update the PIN internally as $PIN_{new} = F(PIN_{old} \oplus G_N)$.

It is suggested that the improved mutual authentication protocol in Section 4 and the improved ownership transfer protocol are used together. Thus, in the protocol in Section 4, the old owner can not get any relationship between IDS and ID of tag because the random number r makes the system obscure. Therefore, property 1) is satisfied. In terms of improved ownership transfer protocol, all the parts in the message are linked together and any element changed will be found by the TTP or tag. So, the new owner can receive and accurately and start performing a normal mutual authentication with the tag. Therefore, property 2) can be satisfied.

4.4 Two-party Ownership Transfer

A two-party ownership transfer solution is a protocol without a TTP can be constructed using improved mutual authentication protocol directly. The setup phase is similar to that in the TTP protocol, the old owner gives the tuple stored (IDS, ID, G_n, G_{n+1}) to the new owner. The online authentication phase is shown in Figure 1 We will not discuss the details again.

5 Conclusion

RFID technology can provide great benefits in several areas and has many applications for both business and individuals. As many technologies, RFID faces also similar security concerns: Authentication, Confidentiality and Availability. For insecure RFID system, the user's privacy will face a great threat. At the same time, it is necessary to avoid expensive cryptographic computations because of low-cost devices and less capability. At INFOCOM 2010, Kulseng et al. gave a lightweight RFID authentication protocol and an ownership transfer protocol which is claimed the most efficient protocols among the existing protocols. However, Kardas et al. point that the protocols have several serious security issues. In this paper, an improved mutual authentication protocol is proposed based on PUF functions and LFSR functions. This paper proves that the proposed protocol is secure against various types of attacks and can solve the problems of the previous works. Furthermore, by satisfying all of the security requirements, the proposed RFID mutual authentication protocol completely solves the privacy and forgery problems. The protocols not only can defeat security attacks but also require small number of gates. Finally, improved ownership transfer is proposed.

Acknowledgments

This work is supported by the Project of The Research of License Plate Recognition System (No.2014j4100201).

References

[1] I. Agudo, R. Rios, and J. Lopez, "A privacy-aware continuous authentication scheme for proximity-based access control," *Computers & Security*, vol. 39, no. 1, pp. 117–126, 2013.
[2] B. Alomair, A. Clark, J. Cuellar, and et al., "Scalable RFID systems: A privacy-preserving protocol with constant-time identification," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1536–1550, 2012.

[3] G. Avoine, C. Lauradoux, and T. Martin, "When compromised readers meet RFID," in *Workshop on Information Security Applications*, vol. LNCS 5932, pp. 36–50, 2009.

[4] L. Bolotnyy and G. Robins, "Physically unclonable function-based security and privacy in RFID systems," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, pp. 211–220, White Plains, NY, March 2007.

[5] M. Chen, W. Luo, Z. Mo, and et al., "An efficient tag search protocol in large-scale RFID systems," in *Proceeding of the 32nd IEEE International Conference Computer Communications*, pp. 899–907, Turin, Italy, April 2013.

[6] S. Kardas, M. Akgun, M. S. Kiraz, and H. Demirci, "Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems," in *Workshop on Lightweight Security Privacy: Devices, Protocols and Applications (LightSec'10)*, pp. 20–25, San Diego, CA, March 2011.

[7] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight mutual authentication and ownership transfer for rfid systems," in *Proceedings IEEE Conference on INFOCOM*, pp. 1–5, San Diego, CA, March 2010.

[8] N. W. Lo and K. H. Yeh, *An Efficient Mutual Authentication Scheme for EPCglobal Class-1 Generation-2 RFID System.* Springer Berlin Heidelberg, 2007.

[9] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic key-updating: Privacy-preserving authentication for RFID systems," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, pp. 13–22, White Plains, NY, March 2007.

[10] D. Moriyama, S. Matsuo, and M. Ohkubo, "Relations among notions of privacy for RFID authentication protocols," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 97, no. 1, pp. 225–235, 2014.

[11] L. M. Ni, Y. Liu, Y. C. Lan, and et al., "LANDMARC: Indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[12] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *First ACM Conference on Wireless Network Security*, pp. 140–147, White Plains, NY, 2008.

[13] Y. Tian, G. Chen, and J. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Communications Letters*, vol. 16, no. 5, pp. 702–705, 2012.

[14] L. Wang, X. Yi, L. V. Chao, and Y. Guo, "Security improvement in authentication protocol for Gen-2 based RFID system," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 157–169, 2011.

[15] J. Zhang, W. Wang, J. Ma, and X. Li, "A novel authentication protocol suitable to EPC class 1 generation 2 RFID system," *Journal of Convergence Information Technology*, vol. 7, no. 3, pp. 259–266, 2012.

Peng-yu Cui received a master's degree in College of Electrical and Control Engineering from North China University Of Technology (China) in June 2011. He is a lecturer in Information Centre in Liao-ning Geology Engineering Vocational College. His current research interest fields include information security and computer application.

An Automatic Alert Unification Method for Heterogeneous Alert Signatures

Ouissem Ben Fredj
The Department of Information Technology, Taif University
P.O.Box: 888, Hawiyah, Taif, Zip Code: 21974, Kingdom of Saudi Arabia
(Email: Ouissem.BenFredj@gmail.com)
(Received Aug. 6, 2015; revised and accepted Nov. 27 & Dec. 8, 2015)

Abstract

Several monitoring systems are usually composed by heterogeneous monitoring sensors. Each sensor raises thousands of alerts to be saved and analyzed in a centralized station. Most of alerts raised by different sensors are almost the same but have various formats and various descriptions. The system administrator must identify manually similar alerts in order to decrease the number of generated alerts and to improve the data quality. This paper proposes an alert unification method that automatically creates meta-alerts from a set of heterogeneous alert sets coming from different security monitoring sensors. Instead of dealing with several sets of alerts, this method allows the administrator to use a unique set of meta-alerts.

Keywords: Data pre-processing, language processing, network security applications, record linkage

1 Introduction

Several systems are usually composed by heterogeneous monitoring sensors. Each sensor has its philosophy, its functional method, and its alert definitions. One important task in such environment is the multisensor data fusion which integrates objects that relate to the same entities from several databases. This task help to improve data quality by producing clean, sanitized, refined, and accurate data ready for fast and simple analysis and good knowledge extraction.

There are several definitions of the architectures of multisensor data fusion system in the literature. Luo and Kay [**?**, **?**, **?**] defined functional roles of multisensor integration and multisensor fusion composed of three-level fusion category. Dasarathy [**?**] proposed an I/O pair-based fusion architecture. [**?**, **?**] provided an introduction to multisensor data fusion based on the architecture of the Joint Directors of Laboratories (JDL) data fusion model [**?**], which was originally developed for military applications. [**?**] gave a review of different models of multisensor data fusion system. [**?**] gave an overview of multisensor fusion techniques relating to different fusion levels of the JDL framework, and discussed the weaknesses and strengths of the approaches in different applications. [**?**] proposed a framework of logical sensor which treats the multisource information in a multisensor system based on the viewpoint of logical software programming. [**?**] imported the JDL processes to the cyber security context.

The most used model is the JDL model [**?**, **?**, **?**] which divides the data fusion process into four levels: Level 1 for object refinement, Level 2 for situation assessment, Level 3 for threat assessment, and Level 4 for process assessment. Level 1 contains processes of data registration, data association, position attribute estimation, and identification. Level 2 fuses the kinematic and temporal characteristics of the data to infer the situation of the environments. Level 3 projects the current situation into the future. In Level 1, the parametric information is combined to achieve refined representations of individual objects. Levels 2 and 3 are often referred to information fusion, while Level 1 is data fusion. Level 4 is an ongoing assessment of other fusion stages to make sure that the data fusion processes are performing in an optimal way.

Bass adapts the JDL model to data fusion in the field of computer security (see Figure **??**) [**?**]. The proposed model adds Level 0 called data refinement. In this level, data acquired from a set of network security sensors (IDSs, network sniffers, application logs), is filtered and calibrated to generate a set of objects. Level 1 correlates all measurements using common spatial and temporal metrics. Level 2 correlates the objects using high level features like their behavior, dependencies, targets, origins, protocols, attack rates. The output of this step is the situational knowledge. Level 3, threat assessment, assesses the situation against known intrusion detection templates and suggests or identifies future threats. Level 4, resource management, analyses the outputs of lower levels (Levels 2, 1, and 0) to define processing priorities to some objects or situations.

This paper proposes a Level 1 method. After gathering alerts from different intrusion detection sensors, the proposed method detects duplicated and related alerts and creates a new set of refined and clean alerts. The method

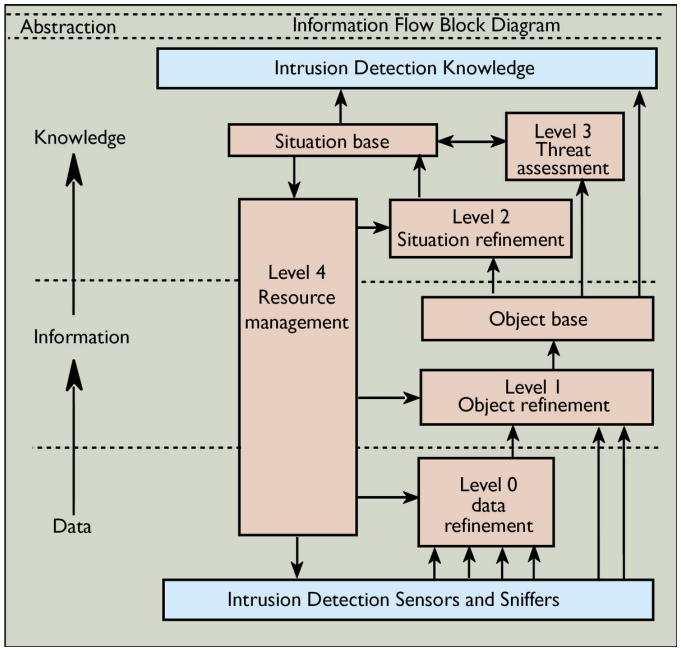


Figure 1: The JDL multisensor data fusion system

makes use of record linkage techniques in order to detect related alerts.

Record linkage aims to identify links between records that refer to the same real-world entities. Most record linkage methods are based on the method of Newcombe and Kennedy in 1962 that was formalized by Fellegi and Sunter in 1969. The literature includes several works in record linkage from various domains [**?**]. Security specialists have used record linkage to remove duplicated alerts raised by Intrusion detection sensors [**?**, **?**, **?**]. It has been also used to identify fraudsters and criminals national security databases [**?**]. The database of historical census data was subject of many investigations that aim to identify links between individuals and even create a complete genealogy tree over a long period of time [**?**, **?**, **?**]. Record linkage was also used to eliminate duplicate records from the result of search engines [**?**, **?**]. Recent researches [**?**, **?**] have employed record linkage methods to determine all bibliographic of an author in large publication databases.

In this paper, we consider a security monitoring system encompassing a distributed IDS (Intrusion Detection System), however, the proposed work could be used in any system that encompasses several heterogeneous monitoring sensors. Usually, large companies deploy several IDS sensors in different locations to gather information about possible threats and attacks. The IDWG (Intrusion Detection Working Group) is a major working group that defined a general distributed IDS architecture [**?**] (see Figure **??**). The E blocks are (Event-boxes) is composed of sensor elements that monitor the target system. The D blocks (Database-boxes) are intended to store information from E blocks for subsequent processing by A and R boxes. The A Blocks (Analysis-boxes) are processing modules for analyzing events and detecting potential hostile behavior, so that some kind of alarm will be generated if necessary. The R blocks (Response-boxes) executes an intrusion reaction.

Figure **??** shows a near-real-time distributed IDS for high-speed networks called ($\phi|\pi$) [**?**]. ($\phi|\pi$) aims to be generic, scalable, and adaptive distributed IDS which acts in real-time depending on traffic, alerts, past traffic, and predicted traffic and alerts.

It is clear that almost all the DIDS architectures include distributed agents/sensors and a global manager which collects information. In order to have a global view of the system, the DIDS must gather events from sensors. The events should be correlated to provide a simple and accurate view to the administrator. The correlation step usually makes use of statistics and data mining in order to improve the monitored site.

The main contributions of this paper are:

- A state of the art of the current architectures of data fusion systems;
- Identification of the relation between data fusion systems and record linkage methods;
- A study on how to use record linkage techniques in the security context;
- A record linkage method that improve data quality of alerts generated by security monitoring sensors;
- A semantic similarity method to compare between alerts;
- An optimization model and solution that solves ambiguity between related alerts.

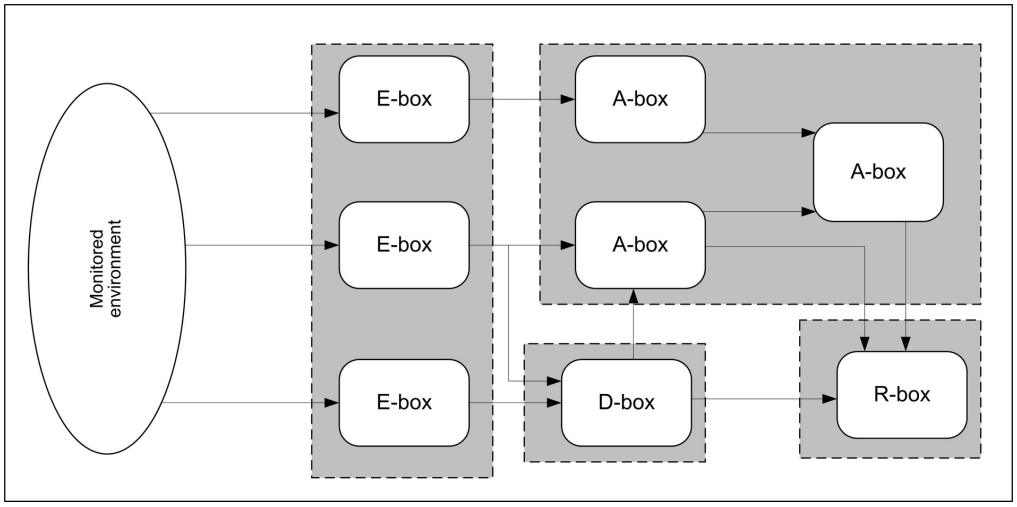


Figure 2: The general IDS architecture as defined by IDWG

The remaining paper is organized as follow: next section defines the problem of record linkage in the context of security monitoring. Section **??** is an overview of the proposed solution called alert unification. Section **??** details the structure of the global algorithm for the alert unification process. Section **??** introduces the semantic-based similarity method to measure the similarity between alerts. Section **??** deals with the introduction of a new alert in the existing system. Section **??** solves the problem of ambiguous related alerts using an original method. Section **??** analyzes the performance of the method, and the last section concludes and discusses the future works.

2 Problem Definition

The alerts defined by a given set of monitoring sensors are different even they refer semantically to the same effective alert. Hence, an alert unification step is required to minimize the total number of generated alerts and to maximize their quality. The alert unification process will combine several alert sets, corresponding to several monitoring sensors, into one unified alert set. Note that this step would be done offline, means that this step would run even before the deployment of the sensors. Thus, sophisticated similarity algorithms could be used to unify alert names. Since each monitoring sensor has its monitoring philosophy and its specific alert format, our method suppose that each generated alert has only one attribute called alert description. Indeed, this simplification makes our method very generic and could be applied with a wide range of monitoring sensors.

Formally, in the current work, we consider a set of sn different sensors monitoring one subject. Each sensor s_i has an alert set $AS_i = A_{(0,i)}, A_{(1,i)}, \ldots, A_{(n_i-1,i)}$ where $A_{(j,i)}$ is the alert number j that belongs to the alert set AS_i and n_i is the size of AS_i i.e. $n_i = |AS_i|$. Each alert set AS_i must verify the following conditions:

- There are no duplicate alerts within the same alert set;
- Each alert in AS_i has a unique identifier ($A_{i,j}$);
- Each alert has an attribute called description.

The result of the unification process is an alert set $U = u_0, u_2, \ldots, u_{q-1}$ containing q unified alerts such that each unified alert u_k is associated with at least one alert $A_{i,j}$. in this case, u_k and $A_{i,j}$ correspond to the same effective alert. u_k could be a unified alert of several alerts from different alert sets (i.e. different sensors). The goal of the unification process is to maximize the number of alerts that a unified alert u_k is associated with. In other words, our goal is to minimize q.

3 Solution Overview

Consider the problem definition above, our proposed solution (see Figure **??**) starts with an indexing step that gives for each alert a unique identifier. During Step 2, each alert is split into word tokens. In order to calculate the similarities between tokens, we propose a semantic similarity approach; given two word tokens, each token has several senses taken from the Wordnet [**?**] database (Step 3). In order to choose the best sense of a given token, we apply a word sense disambiguation method called the Mickeal Lest algorithm [**?**] (Step 4). Then, a similarity value is computed between the selected senses of both tokens using the Resnik method [**?**] (Step 5). The Resnik similarities between tokens senses create a similarity matrix between the tokens of two alerts. The similarity matrix is transformed to an optimization problem aiming to maximize the overall similarities between the tokens of both alerts in order to choose the best matching

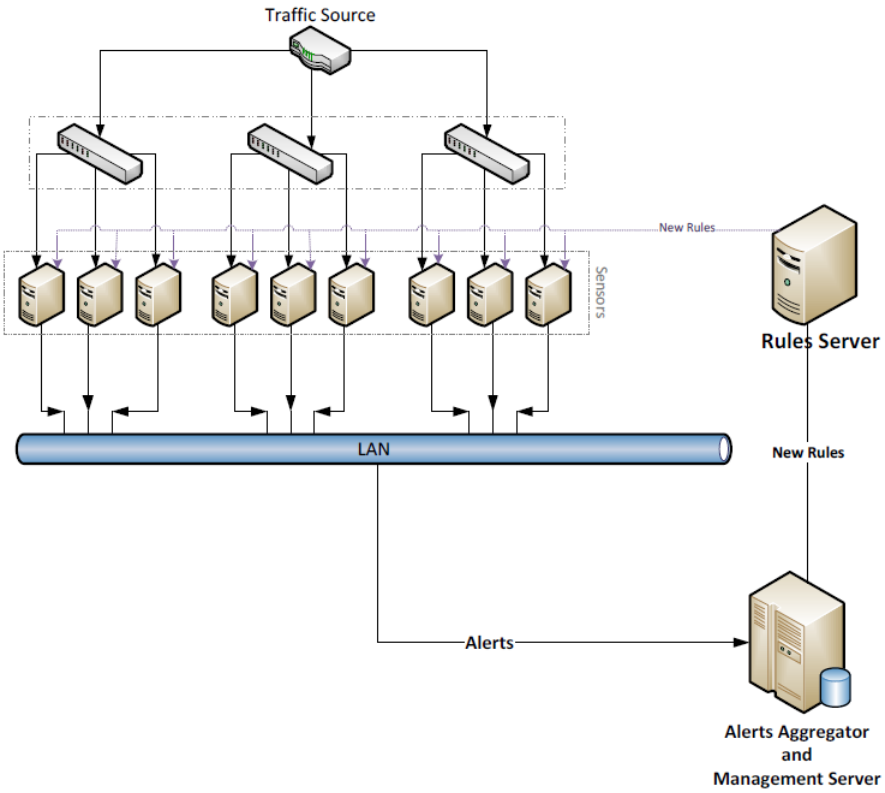


Figure 3: The global architecture of a phi DIDS

between tokens. The optimization problem is solved using the polynomial Hungarian method [**?**] (Step 6). The output of the later step is the similarity value between two alerts (Step 7). The previous steps are repeated to compute the similarity between each couple of alerts. In order to minimize the complexity of the global algorithm, the unified alert set U is created incrementally; each new alert is compared to the existing unified alerts (using the previous 7 steps). The result of the previous comparison is one of the following three cases:

- The new alert is not similar to any of the unified alerts (using a high similarity threshold). In this case the new alert become a new unified alert and added to the unified alert set U;
- The new alert is similar to only one unified alert. In this case the new alert is considered as a duplicate of the unified alert;
- The new alert is similar to several unified alerts. This case raises an ambiguity problem between the alerts. However, the new alert must be a duplicate of only one unified alert. The ambiguity problem is solved by an optimization method called Hopcroft-Karp algorithm as explained in Section **??**.

4 The Alert Unification

In this section, the main steps of the alert unification are detailed. Given sn different IDS sensors. Each sensor s_i has an alert set $AS_i = A_{(0,i)}, A_{(1,i)}, \ldots, A_{(n_{i-1},i)}$ where $A_{(j,i)}$ is the alert number j that belongs to the alert set AS_i and n_i is the size of AS_i i.e. $n_i = |AS_i|$. The result of the unification process is an alert set $U = u_0, u_2, \ldots, u_{q-1}$ such that each u_k is associated with at least one alert $A_{i,j}$. in this case, u_k and $A_{i,j}$ correspond to the same alert. u_k could be the unified alert of several alerts from different alert sets AS_i (i.e. different sensors). Let UM is the unification matrix. Hence, UM_k is the unification vector of the unified alert u_k. $UM_{k,p}$ is the alert number from the alert set S_p that corresponds to the unified alert u_k. For example $UM_{5,2} = 7$ means that the fifth alert of U corresponds to the seventh alert of the alert set of the IDS sensor 2. $UM_{5,3} = -1$ means that the unified alert u_5 has not any corresponding alert with the alert set of the sensor 3.

$$UM_{k,j} = \begin{cases} i & if \quad u_k \approx AS_{i,j} \\ -1 & if \quad \forall i, \quad u_k \not\approx AS_{i,j} \end{cases}$$

For two alerts u and v, $u \approx v$ means that the alert u and v relate to the same alert even they differ in their respective descriptions, and $u \not\approx v$ means that the alert u and v are different. Note that UM has q (i.e. $|U|$) rows and sn (the number of different IDS sensors) columns. Also, q is initially unknown. $q \in [\max_{i\in[0,sn-1]} n_i, \sum_{i=0}^{sn-1} n_i]$

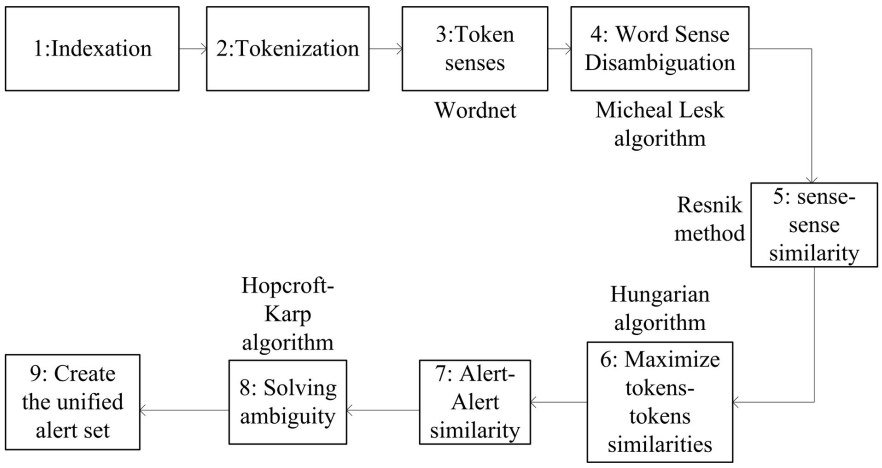


Figure 4: the main steps of the alert unification process

The goal of the unification process is to maximize the number of alerts that a unified alert u_k is associated with. Thus, minimizing the (-1)s in the unification matrix UM. The goal of the unification process can be viewed also as minimizing q. The proposed approach builds UM column by column. The steps are summarized in the following:

Inputs:

- sn alert sets: $AS_0, \cdots, AS_{sn-1}$;
- Each alert set AS_i is composed by n_i alerts: $S_i = A_{i,0}, A_{i,1}, \cdots, A_{i,n_{i-1}}$. It is preferable that the alert sets are decreasingly ordered according to their size, $n1 \geq n2 \geq \cdots \geq n_{sn-1}$.

Outputs:

- The unified alert set $U = u_0, u_2, \cdots, u_{q-1}$. q is the number of unified alerts.
- The unification matrix UM.

Procedures:

1) Clone AS_1 to U. U is initialized with the first alert set AS_1. Thus,
   - $U := AS_1$
   - $q := |AS_1|$
   - $UM_{i,1} := i$, for each $i \in [0, |AS_1| - 1]$. Each alert of AS_1 is similar with itself.
   - $j := 2$
2) Build the similarity matrix SS_{U,S_j} between each couple of alert sets U and AS_j. $SS_{U,S_j}[a, b]$ is the similarity rating between the alerts u_a and $A_{j,b}$. We assume that all the scores of the similarity matrix are in the range $[0, 1]$, which means that if the score gets a maximum value (equal to 1) then the two alerts are absolutely similar. The algorithm used to build the similarity matrices is explained below (see Section **??**).
3) Set a hard similarity threshold ST for example (0.9) which guarantees the similarity between the alerts. In other words, if $SS_{U,S_j}[a, b] > ST$, then we assume that the alerts u_a and $A_{j,b}$ could be similar.
4) Set to zero the values of the similarity matrix SS_{U,S_j} which are less than ST. i.e. $SS_{U,S_j}[a, b] = 0$ if $SS_{U,S_j}[a, b] < ST$. $SS_{U,S_j}[a, b] = 0$ means that the alerts u_a and $A_{j,b}$ could not correspond to the same alert.
5) if $SS_{U,S_j}[a, b] > ST$ and $\nexists\ c$ such as $SS_{U,S_j}[a, c] > ST$ then $UM_{a,j} = b$. i.e. if the alert u_a is similar to only one alert $A_{j,b}$, then we assume that the unified alert of $A_{j,b}$ is u_a:
   - $UM_{a,i} := 0$, for each $i \in [0, sn - 1]$, the new unified alert has not any corresponding alert except $A_{j,b}$.
   - $UM_{a,j} = b$
6) if $SS_{U,S_j}[a, b] = 0$ for all $b \in [0, |AS_j| - 1]$ i.e. the alert $A_{j,b}$ has not any corresponding unified alert, then, add $A_{j,b}$ as a new unified alert:
   - $q := q + 1$, increment the number of unified alerts
   - $u_q := A_{j,b}$, the new unified alert corresponding to $A_{j,b}$
   - $UM_{q,i} := 0$, for each $i \in [0, sn - 1]$, the new unified alert has not any corresponding alert except $A_{j,b}$.
   - $UM_{q,j} := b$
7) if $SS_{U,S_j}[a, b] > ST$ and $\exists\ c$ such as $SS_{U,S_j}[a, c] > ST$. i.e. the alert u_a is a unified alert of several eventual alerts from the same alert set S_j. This is problematic because several alerts from S_j are ambiguous while u_a must correspond to only one alert from the set S_j. Hence, we propose to find a support to choose the most appropriate alert. See details of the

approach in Section **??**. Let the alert $A_{j,b'}$ the supported alert. Thus, the unified alert is voted to correspond to $A_{j,b'}$:

- $UM_{a,i} := 0$, for each $i \in [0, sn-1]$, the new unified alert has not any corresponding alert except $A_{j,b'}$.
- $UM_{a,j} := b'$

8) If $j < sn$ then increment $j (j := j+1)$ and go to Step 2.

5 Similarity Measure Between Alerts

As introduced in the last section, the similarity between alert sets requires the similarity matrix SS that summarizes the similarities between alerts. This step is the major step of the unification process since it may affect the overall accuracy of the whole process. Optimistic similarity measure may lead to incorrect fusion between different alerts giving the administrator a false view of the system. Pessimistic similarity measure supports dissimilarity between alerts which may increase the number of unified alerts; each one corresponds to a few effective alerts. The worst case is that each unified alert corresponds to only one effective alert. In our approach, we propose to take advantage from the semantic similarity approaches in order to compare the descriptions of the alerts that have different names. Thus, we compare senses and not words. We propose an approach based on semantic similarity. Given two sentences, semantic similarity gives a score that reflects the semantic relation between the meanings of them. The semantic similarity algorithm may take advantage from the WordNet [**?**] semantic dictionary. WordNet is a lexical database of English. English words (Nouns, adjectives, verbs and adverbs) are grouped into sets of synsets which are cognitive synonyms. Each synset expresses a distinct concept. Wordnet graph's edges are the relations between synsets. Relations are set up by means of lexical relations and conceptual-semantic that results in a network of related words and concepts. These relations vary based on the type of word. For the current work, we limit the considered word types to nouns and verbs and we limit the relations to the following:

For nouns: hypernyms, hyponyms, holonym, and meronym.

For verbs: hypernym, troponym, entailment, and coordinate terms.

The steps involved in the semantic similarity are the following:

Inputs:

- Let X and Y two alerts;
- DX the sentence that correspond to the description of the alert X;
- DY the sentence that correspond to the description of the alert Y.

Output:

- Similarity score $sim(X, Y)$ between X and Y

Procedures:

1) If $X = Y$ then $sim(X, Y) = 1$; exit!
2) Tokenization of DX and DY;
   - Remove the stop words;
   - Remove the articles;
   - Split the sentences into a list of words (tokens);
   - We denote m to be the number of tokens of DX, and n to be the number of tokens of DY.
3) Identify the eventual senses of each token using Wordnet.
4) Identify the best sense of each token. This step takes advantage of the Word Sense Disambiguation (WSD) algorithms to identify the most appropriate sense of a word used in a given sentence, when the word has multiple senses (polysemy).
5) Similarity $sim(t_i, t_j)$ between each couple of two senses (t_i, t_j) where ti is the i^{th} sense in the description of the first alert and t_j is the j^{th} sense in the description of the second alert. The step makes use of the Resnik Method as explained later.
6) Similarity of tokens without senses. If a word does not exist in the dictionary, such that in the case of abbreviations and acronyms, we use the following binary similarity measure:

$$sim(t_1, t_2) = \begin{cases} 1 & if \quad t_1 = t_2 \\ 0 & Otherwise \end{cases}$$

7) Build the similarity matrix between all the tokens of both alerts.
8) Identify the best similarity matching: for each token from X, identify a token from Y that maximizes the similarities between all token of X and Y.
9) Compute the similarity between X and Y: $\sim (X, Y)$.

The first two steps of the algorithm are easy. Regarding Step 3, there are three categories of the proposed techniques; the dictionary-based methods, the completely unsupervised methods, and the supervised machine learning methods based on a corpus of manually sense-annotated examples. An example of this first approach is the Micheal Lesk algorithm [**?**, **?**]. The objective of the algorithm is to count the number of words that are shared between two glosses (definitions). The more overlapping the words, the more related the senses are. Given a word to be disambiguated, the dictionary

definition of each of its senses is compared to the glosses of every other word in the phrase. The sense whose gloss shares the largest number of words with the glosses of the other words is selected. The method begins anew for each word and does not utilize previously assigned senses. Formally, given two words w_1 and w_2, the score of each pair of word senses $S_1 \in Senses(w_1)$ and $S_2 \in Senses(w_2)$:

$$score_{Lesk}(S_1, S_2) = |gloss(S_1) \cap gloss(S_2)|$$

where $gloss(S_i)$ is the set of words in the definition of sense S_i of word w_i. The senses that maximize the score formula are assigned to the respective words. The WordNet glosses (definitions) might be used as dictionary definition database. The following pseudo code describes the original Lesk algorithm [**?**].

Algorithm 1 Lesk algorithm

```
1: Begin
2: for every word w[i] in the alert do
3:    let best_score = 0
4:    let best_sense = null
5:    for every sense sense[j] of w[i] do
6:       let score = 0
7:       for every other word w[k] in the alert, k! = I do
8:          for every sense sense[l] of w[k] do
9:             score = score + number of words that occur
               in the gloss of both sense[j] and sense[l]
10:         end for
11:      end for
12:      if score > best_score then
13:         best_score = score
14:         best_sense = w[i]
15:      end if
16:   end for
17:   if best_score > 0 then
18:      return w[i] the word w[i] has is the best sense.
19:   else
20:      return nil
21:   end if
22: end for
23: End
```

The Lesk algorithm requires the calculation of $|Senses(w_1)|.|Senses(w_2)|$ gloss overlaps. In a context of n words, we need to compute $\prod_{i=1}^{n} |Senses(wi)|$ overlaps, which require an exponential number of steps. However, this is not problematic since the unification process is offline. The goal of Step 4 is to compute the similarity $sim(t_i, t_j)$ between each sense couple (t_i, t_j) where t_i is the i^{th} sense in the description of the first alert and t_j is the j^{th} sense in the description of the second alert. In WordNet, if a word has more than one sense, it will appear in multiple synsets at various locations in the graph. WordNet defines relations between synsets and relations between word senses. Figure **??** shows an example of the graph generated by WordNet including nouns and verbs.

Furthermore, we take advantages from the Resnik approach [**?**] to compute similarity between senses. The approach is based on the notion of information content. The method assumes that one criterion of similarity between two concepts is "the extent to which they share information in common", which in an IS-A taxonomy can be computed by examining the relative position of the most-specific concept that subsumes them both. An information content $IC(c)$ of a concept c is the probability $p(c)$ of finding an instance of the concept c in a given corpus. Following the standard argumentation of information theory [**?**], the information content of a concept c can be quantified as negative the log likelihood:

$$IC(c) = -logp(c)$$

IC has a lower value for the more abstract a concept.

Regarding the similarity between concepts, Resnik stated that, the more information two concepts share in common, the more similar they are. That means that, given the taxonomy graph, the shorter the path from one node to another, the more similar they are. Thus, the Resnik method only considers the information content (IC) of lowest common subsumer (LCS). A LCS is a concept in taxonomy (WordNet in our case), which has the shortest distance from the two given concepts. That is, the LCS of two synsets is the most specific common subsumer of the two synsets (most specific ancestor node). The similarity between two concepts $c1$ and $c2$ is then defined as:

$$sim(c1, c2) = -logp(lcs(c1, c2))$$

Note that the Resnik method assume that a root node of the taxonomy graphs exists which is not always true because the different top nodes of each word type taxonomy are not joined. For that, we create a root node that joins all the top node of the different taxonomies. Then, a path will certainly exist between any two concepts.

The probability $p(c)$ of finding an instance of the concept c in a given corpus is calculated as the following:

$$p(c) = \frac{\sum_{n \in W(c)} count(n)}{N}$$

where $W(c)$ is the set of words in the WordNet corpus whose senses are subsumed by concept c, and N is the total number of word (noun) tokens in the corpus that are also present in WordNet. Resnik used the Brown Corpus of American English as the corpus.

Now, given two words w_1 and w_2, the similarity between them is computed as follow:

$$sim(w_1, w_2) = \max_{c1 \in s(w_1), c2 \in s(w_2)} sim(c1, c2)$$

where $s(w)$ is the set of concepts in the taxonomy that are senses of word w [**?**]. That is, the similarity between two words is equal to that of the most-related pair of concepts that they denote.

Step 6 aims to build a matrix that summarizes the similarity between the tokens of the two alerts. The result is the similarity relative matrix $R[m, n]$ of each pair of token senses, $R[i, j]$ is the similarity between the token

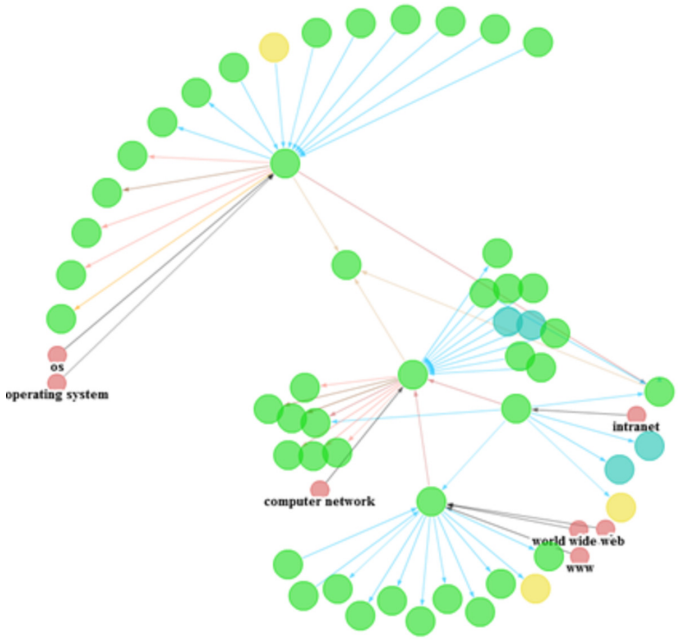


Figure 5: An example of wordNet taxonomy that includes the word types noun and verb and the relation between them. Image generated by Wordnet Editor [**?**]

sense i of the first alert and the token sense j of the second alert. The matrix R is the result of similarity measures of Steps 4 and 5.

The Step 7 is little bit tricky, the goal is to identify the best similarity matching between all the tokens of both alerts X and Y in a way that maximizes the overall similarity scores. Computing semantic similarity between two sentences could be formulated as the problem of computing the maximum total matching weight of a bipartite graph. The nodes of the graph are the tokens of X and Y and the edges are $R[i,j]$, the weight of the edge connecting from token $i \in X$ to token $j \in Y$. Let G denotes such a graph. The graph is often non balanced bipartite graph since the number of tokens in the first alert is often not equal to the number of tokens of the second alert ($|X| \neq |Y|$). Also, the graph is a complete bipartite graph because for any two vertices $i \in X, j \in Y$, ij is an edge in the graph G. The goal of the maximum total matching weight of a bipartite graph is to find a matching M in G which maximizes, over all possible matching, the total amount, $(M) = \sum_{ij \in M} R[i,j]$, of cost consummated by M. The maximum total matching weight of a bipartite graph could be formulated as an assignment problem [**?**].

The assignment problem consists on a number of agents and a number of tasks. Any agent can be assigned to perform any task for a certain cost. The goal is to perform all tasks by assigning exactly one agent to each task in such a way that the total cost is minimized. The main difference between the maximum total matching weight of a bipartite graph and the assignment problem is that the first problem tries to find the matching that maximizes the cost and the last problem tries to find the optimal assignment that minimizes the cost. One famous algorithm that could solve the assignment problem, in polynomial time of the number of nodes of X, is the Hungarian algorithm [**?**]. However, the method requires two firm conditions:

$$\begin{cases} linear \quad assignment \quad problem : |X| = |Y| & (1) \\ the \quad goal \quad is \quad to \quad minimize \quad the \quad overall \quad cost & (2) \end{cases}$$

The first condition imposes that the number of agents and tasks are equal which means that the number of token in X must be equal to the number of tokens in the second alert. This problem could be overcome by adding dummy tokens in the small token set with minimum cost 0 (see Figure **??**).

Whereas, the second condition impose a minimization problem of the cost which is in contrast with our goal to maximize the overall similarity, two hints exist that allow using the Hungarian method for a maximization problem [**?**]. The first is to multiply the matrix R by -1. The second method suggest to replace each element R_{ij} by $max(R) - R_{ij}$. Where $max(R)$ is the maximum value that exist in R. Since the elements of R are probabilities, the transformation will be reduced to $R_{ij} = 1 - R_{ij}$.

The last Step 8 allows to compute the total similarity score $sim(X,Y)$. This is done by combining the match results of the previous step into a global similarity score for both alerts X and Y:

$$sim(X,Y) = \frac{2 * \sum_{x \in DX, y \in DY}^{min(|DX|,|DY|)} sim(x,y)}{|DX| + |DY|}$$

This global similarity is deduced by dividing the sum of similarity scores of all match tokens of both sentences X and Y (Step 7) by the total number of both tokens.

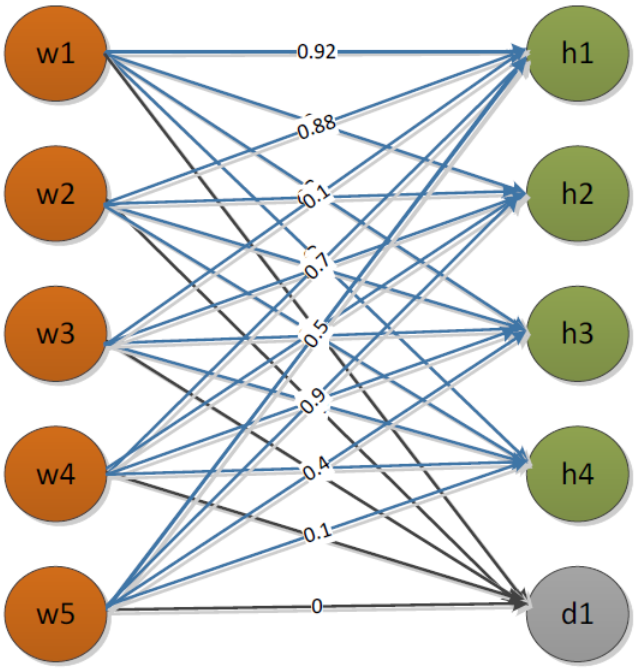


Figure 6: Best matching problem. wi are the token senses of the first alert W, h_j are the token senses of the second alert H. Since $|W| < |H|$, a dummy node d_1 is added in H. The edges oriented to d_1 have cost 0.

6 Classification of New Alert Signatures

When a new alert signature A_a is added to the alert set of a given alert set S_i. The alert A_a is compared to the list of unified alerts U except the alerts that have a corresponding alert in S_i. This is because two alerts in S_i could not be similar and indeed they could not have the same unified alert.

If the alert A_a is similar (the similarity measure is above the threshold ST defined in Step 3 of Section **??**) to one or more unified alerts, then let u_b the unified alert that has the maximum similarity measure. The unification matrix is updated to set u_b as the unified alert corresponding to A_a. When a new IDS type is added to the architecture of the DIDS, the algorithm of Section **??** must be applied to the new alert set of the new IDS type. The Algorithm must take as input the unified alert set U and the new alert set.

7 Solving Alert Similarity Ambiguity

In Section **??**, when u_a is a unified alert of several eventual alerts from the same alert set S_j, one cannot choose the alert from S_j with highest similarity because the difference between the selected alerts of S_j is minimal. An incorrect alert classification could lead to severe consequences. However, one of the selected alerts of S_j is appropriate since their similarity values are above the hard threshold.

The two alert sets and the similarity relation between their alerts can be described as a bipartite graph $G = (U, V, E)$. U and V denote the two partition nodes and E denotes the edges between U and V. In our case, U is the unified alerts and V is the alerts of S_j. E is the similarity relations that exists between the alerts. G is bipartite since there is no edge connecting two nodes from the same partition. That is, the similarity relation is defined between the alerts of U and the alerts of V. Formally, for every edge $uv \in E$, $u \in U$ and $v \in V$.

Figure **??** shows an example of alert similarity ambiguity formulated as a bipartite graph. The unified alert u_2 is similar to a_1, a_2 and a_3, whereas, the alert a_2 is similar to both alerts u_2 and u_3. To resolve the ambiguity, each unified alert must correspond to only one alert. A good approach is to select the maximum edges possible in a way that each unified alert is similar to only one alert in S_j.

The ambiguity resolution problem could be reduced to finding a set of pairwise non-adjacent edges denoted M. That is, no two edges share a common node. M must contain the largest possible number of edges. This problem is called the maximum matching graph problem or the maximum independent edge set problem.

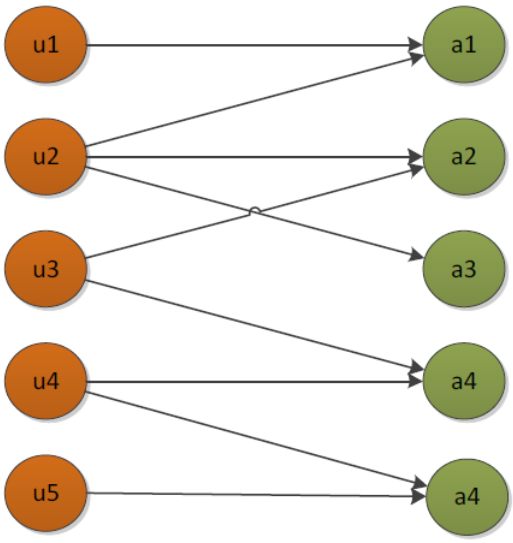


Figure 7: The alert similarity ambiguity is formulated as a bipartite graph. The left set contains the unified alerts and the right set contains the ambiguous alerts of S_j.

The Hopcroft-Karp algorithm [**?**] can solve the maximum matching problem for bipartite graph as shown in Algorithm **??**.

The operator $\ominus$ used for two sets is the symmetric difference. Once the maximum matching M is found, the remaining unmatched alerts ru will be considered as new unified alerts. Formally, $ru = i : i \in Vandui \notin M$.

Algorithm 2 The Hopcroft-Karp algorithm

Input: − $G = (U, V, E)$ a bipartite graph
Output: − The maximum matching M of G
Begin
$M = \emptyset, U_0 = U, V_0 = V$
while $U_0 \neq \emptyset$ **do**
  $L_0 = U_0, k* := k := 0$
  while $L_k \neq \emptyset$ **do**
    construct a layered graph:
    for all $i \in L_k$ **do**
      $N_i = j : ij \in E \setminus M, j \in L_1 \cup L_2 \cup \cdots L_{k-1}$
    end for
    $L_{k+1} = \cup_{i \in L_k} N_i$
    if $L_{k+1} = \emptyset$ **then**
      return M
    end if
    if $L_{k+1} \cap V_0 \neq \emptyset$ **then**
      $k^* := k + 1$
      $L_{k+2} := \emptyset$
    else
      $L_{k+2} := i' : i'j \in M, j \in L_{k+1}$
    end if
    $k := k + 2$
  end while
  delete all vertices in $L_{k^*} \setminus V_0$
  mark all remaining vertices as unscanned
  $k := 1$: path counter
  while $L_0 \neq \emptyset$ **do**
    $x_0 = i \in L_0, L_0 = L_0 \setminus i, l := 0$
    while $l \geq 0$ **do**
      while x_l has unscanned neighbor in L_{l+1} **do**
        Choose unscanned neighbor x_{l+1}
        Mark x_{l+1} as scanned
        $l := l + 1$
        if $l = k^*$ **then**
          $P_k = (x_0, x_1, \ldots, x_{k^*}); k = k + 1$

        end if
      end while
      if $l < k^*$ **then**
        $l := l - 1$
      else
        $l := -1$
      end if
    end while
  end while
  $P := (P_1, P_2, \ldots, P_{k-1})$
  $M = M \ominus P$
  Update U_0, V_0 of unmatched vertices
end while
return M
End

8 Performance Analysis

The complexity of the unification process is not a big issue since the process will run offline. However, this section will show that the time complexity of the process is acceptable. Suppose that there are n monitoring sensors, each sensor has m alerts, each alert has an average of p tokens, and each token has an average of q senses. The complexity times of the alert unification is: $c = c(indexation) + c(tokenization) + c(Lesk) + c(Resnik) + c(Hungarian) + c(Hopcroft - Karp)$.

The complexity of the indexation step is proportional to the number of alerts ($n*m$). The tokenization performs an average of p steps for each alert which requires a linear time complexity. Lesk algorithm is applied for each alert and requires qp steps for each alert ($O(n * m * p * qp)$). The Hungarian algorithm runs in polynomial time $O(p4)$ for each alert, and $O(n * m * p4)$ for all the alerts. The Hopcroft-Karp will run occasionally when the alert sets include ambiguities. The time complexity of the algorithm is $O(|E|\sqrt{|V|})$ in the worst case where the E is the set of the edges and V is the set of nodes of the bipartite graph.

The time complexity of Lesk algorithm is the biggest complexity $O(n * m * p * qp)$. However, in real world scenarios, the alerts has about ten tokens, each token has an average of 3 senses [**?**]. The time complexity of Lesk algorithm is good enough. The overall complexity of the unification process is polynomial.

9 Conclusions and Future Works

The paper has stated the importance and the current uses of the distributed monitoring systems and their current challenges. The problem of heterogeneity of alerts raised by different sensors is widely discussed in the literature. We introduced an approach of an automatic alert unification system that deals with heterogeneous alert signatures. The result is a set of unified alerts. The method used advanced linguistic models and optimization models in order to perform a semantic comparison between alerts. The method resolved also the problem of ambiguity of very similar alerts using an efficient optimization method.

A number of works still opened for future investigations. Regarding the current work, the performance results are acceptable for an offline alert unification. However, there are several monitoring systems that generate unpredictable alert signatures and the alert unification would be done on the fly. The current unification method should be adapted to such scenarios. Second, the paper solves the problem raised by the introduction of a new alert and how to know if it belongs to an existing unified alert. But what about feeding several new alerts at the same time? Is there any method that is faster that processing the new alerts one by one?

References

[1] L. Antonie, K. Inwood, D. J. Lizotte, J. A. Ross, "Tracking people over time in 19th century Canada

for longitudinal analysis," *Machine Learning*, vol. 95, no. 1, pp. 129–146, 2014.

[2] S. Banerjee and T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet," in *Computational Linguistics and Intelligent Text Processing*, pp. 136–145, Springer Berlin Heidelberg, 2002.

[3] T. Bass, "Intrusion detection systems and multisensor data fusion", *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[4] M. Bateni, A. Baraani, and A. Ghorbani, "Using artificial immune system and fuzzy logic for alert correlation," *International Journal of Network Security*, vol. 15. no. 3, pp. 190–204, 2013.

[5] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, Social for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[6] P. Christen, "Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection," *Data-Centric Systems and Applications*, Springer, 2012.

[7] B. V. Dasarathy, "Sensor fusion potential exploitation Innovative architectures and illustrative applications," *Proceedings of IEEE*, vol. 85, no. 1, pp. 24–38, Jan. 1997.

[8] H. Debar, D. Curry, B. Feinstein, *The Intrusion Detection Message Exchange Format (IDMEF)*, Internet Engineering Task Force, IETF RFC 4765 (Experimental), 2007.

[9] V. R. Dondeti and H. Emmons, "Max-min matching problems with multiple assignments," *Journal of Optimization Theory and Applications*, vol. 91, no. 2, pp. 491–511, Nov. 1996.

[10] W. Elmenreich, "A review on system architectures for sensor fusion applications," in *Software Technologies for Embedded and Ubiquitous Systems*, pp. 547–559, 2007.

[11] A. A. Ferreira, et al., "Self-training author name disambiguation for information scarce scenarios," *Journal of the Association for Information Science and Technology*, vol. 65, no. 6, pp. 1257–1278, 2014.

[12] Z. Fu, M. Boot, P. Christen, and J. Zhou, "Automatic record linkage of individuals and households in historical census data," *International Journal of Humanities and Arts Computing*, vol. 8, no. 2, pp. 204–225, 2014.

[13] P. Garcia-Teodoro, J. E. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & SecurityZ*, vol. 28, no. 1, pp.18-28, 2009.

[14] N. A. Giacobe, "Application of the JDL data fusion process model for cyber security," in *SPIE Defense, Security, and Sensing*, pp. 77100R, International Society for Optics and Photonics, 2010.

[15] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Boston, MA: Artech House, 1992.

[16] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of IEEE*, vol. 85, pp. 6–23, Jan. 1997.

[17] T. C. Henderson and E. Shilcrat, "Logical sensor systems," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 169–193, 1984.

[18] J. E. Hopcroft and R. M. Karp, "An n5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal of Computing*, pp.225–231, 1973.

[19] J. Jonas and J. Harper, *Effective Counterterrorism and the Limited Role of Predictive Datamining*, Cato Institute, 2006.

[20] P. Kabiri, A. A. Ghorbani, "A rule-based temporal alert correlation system," *International Journal of Network Security*, vol. 5. no. 1, pp. 66–72, 2007.

[21] G. Kalpana, R. P. Kumar, and T. Ravi, "Classifier based duplicate record elimination for query results from web databases," in *IEEE Trendz in Information Sciences & Computing*, pp. 50–53, 2010.

[22] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83-97, 1955.

[23] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice scream cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation*, pp. 24–26, 1986.

[24] R. C. Luo and M. G. Kay, "A tutorial on multisensor integration and fusion," in *Proceedings of 16th Anual Conference on IEEE Industrial Electronics*, pp. 707–722, 1990.

[25] R. C. Luo and M. G. Kay, "Multisensor fusion and integration in intelligent systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 901–931, Sep./Oct. 1989.

[26] R. C. Luo and M. G. Kay, *Multisensor Integration and Fusion for Intelligent Machines and Systems*, Norwood, MA: Ablex Publishing, 1995.

[27] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp.448–453, 1995.

[28] S. Ross, *A First Course in Probability*, Macmillan, 1976.

[29] C. Rudin, and K. L. Wagstaff, "Machine learning for science and society," *Machine Learning*, vol. 95, no. 1, pp. 1–9, 2014.

[30] H. Sallay, M. Rouached, A. Ammar, O. B. Fredj, K. Al-Shalfan, and M. B. Saad, "Wild-inspired intrusion detection system framework for high speed networks ($\phi|\pi$) IDS framework," in *Privacy Solutions and Security Frameworks in Information Protection*, pp. 241, 2012.

[31] C. Schulz, et al., "Exploiting citation networks for large-scale author name disambiguation," *EPJ Data Science*, vol. 3, no. 1, pp. 1–14, 2014.

[32] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE*

Transactions on Knowledge and Data Engineering, vol. 18, no. 12, pp. 1696–1710, Dec. 2006.

[33] W. Su, J. Wang, and F. H. Lochovsky, "Record matching over query results from multiple web databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 4, pp. 578–589, 2010.

[34] U.S. Dept. Defense, *Data Fusion Lexicon*, Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, 1991.

[35] *WordNet is a Large Lexical Database of English*, 2016. (`http://wordnet.princeton.edu/`)

[36] *Wordnet Editor*, 2016. (`http://wordventure.eti.pg.gda.pl/wne.html`)

[37] WordNet statistics, Aug. 2015. (`https://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html`)

Ouissem Ben Fredj received the BE degree in computer science from the University Manar II, Tunisia in 2002. He obtained the MS in computer science from University of Henri Poincare, France in 2003. He Obtained the PhD degree in computer science from University of Val d'Essonnes, France in 2007. He is currently an assistant professor of computer science at Taif University, Saudi Arabia. His research interests include Vulnerability Assessment, Network Security, and Distributed Systems.

A Publicly Verifiable Secret Sharing Scheme Based on Multilinear Diffie-Hellman Assumption

Qiao Peng[1], Youliang Tian[1,2]
(Corresponding author: Youliang Tian)
College of Science, Guizhou University[1]
Institute of Cryptography and Data Security, Guizhou University[2]
Guiyang 550025, China
(Email: youliangtian@163.com)[2]
(Received Oct. 22, 2015; revised and accepted Dec. 7 & Dec. 17, 2015)

Abstract

Using multiple linear of multilinear map, we propose a simple, non-interactive and effective publicly verifiable secret sharing (PVSS) scheme based on multilinear Diffie-Hellman assumption (MDH). Up to now, the publicly verifiable of secret sharing is still an issue. In this paper, we set the sharing secret is a multiple linear pairing, we apply the multiple linear property of multilinear map for the shares authentication to achieve publicly verifiability of secret sharing. What's more, the batch verification technique is used to reduce the computational overhead at share verification phase. Compared with the existing programs, this scheme has improved communication efficiency under the same security level and it can meet those high efficiency and security of the communication requirements of the application scenarios. In addition, we apply our PVSS scheme to electronic voting skillfully. At last, the performance analysis results show the publicly verifiability, security and practicality of our scheme in the random oracle and under MDH assumption.

Keywords: Electronic voting, multilinear map, multilinear Diffie-Hellman assumption, publicly verifiable secret sharing

1 Introduction

Secret sharing is an important research content of modern cryptography, it is a method of increasing the security of cryptography system. The earliest secret sharing schemes were proposed by Shamir [20] and Blakey [2] in 1979. Shamir's (t, n) threshold secret sharing scheme is based on polynomial interpolation in a finite field. In Shamir's scheme, the secret is able to be reconstructed by t or more participants at secret reconstruction phase, while any subset of t-1 or less participants has no information about the secret. Threshold secret sharing [1, 10] has remarkable effect on cryptography due to its effective and applicability. However, it still has the following drawbacks:

1) Unable to identify the honesty of the dealer;

2) Unable to detect dishonest participants and prevent cheating behavior.

In order to prevent malicious behavior of the dealer and participants, a new type of secret sharing scheme was first proposed by Feldman [11], called Verifiable Secret Sharing (VSS) schemes, which solved the security issues mentioned above. However, it also pledges that the participants only verify their own shares but cannot verify the other participants received shares. VSS scheme such as all require the availability of private channel from the dealer to each of the participants individually, but the communication over the private channel of VSS scheme is obviously not publicly verifiable.

However, in publicly verifiable secret sharing (PVSS) scheme, the dealer broadcasts information to the participants without needing to maintain a private channel, which avoids the interaction between the dealer and the participants, as well as the interaction among the participants. The notion of PVSS was first introduced by Stadler [21] in public key setting. PVSS scheme with the objective that anyone, not just the participants, can verify whether the distributed shares are valid without revealing any information about the secret at the secret distribution phase and whether each participant releases the correct share at the secret reconstruction phase. Moreover, Stadler expressed the main goal of threshold secret sharing scheme was that each authorized subset of the access structure could reconstruct the secret. The PVSS scheme reduces the overhead of communication and safeguards the security of the scheme because it does not require private channels. In view of these advantages, Schoenmakers [19] proposed a simple PVSS scheme based on discrete logarithm problem and gave its applications in electronic voting systems and key escrow. Later, some publicly ver-

ifiable secret sharing schemes based on traditional public-key systems were proposed [6, 7, 26]. Although PVSS plays a powerful role in threshold cryptography, the security of this kind of scheme was either based on the integer factoring problem or the discrete logarithm problem. Until 1993, Menezes et al. [16] presented the Weil pairing, which was defined on an elliptic curve and could be used to solve the decision Diffie-Hellman (DDH) problem effectively. Subsequently, many pairing-based secret sharing schemes were proposed [3, 23, 24]. For example, WU and TSENG [25] proposed the first pairing-based PVSS scheme in 2011, they had showed the security of their PVSS scheme under the bilinear Diffie-Hellman assumption, but the computation overhead of their scheme was considerable, especially in the share verification phase.

Recently, multilinear map has received extensive attentions from cryptographic researchers, which has been applied to public key cryptography [8, 12] successfully. In [13], Garg et al. presented a public and secure attribute-based signcryption scheme based on multilinear map, this signcryption scheme gave the foundation method of carrying out secure communication in social network. In 2009, on the basis of multilinear map, by using multiple linear pairing, Ruckert et al. [18] have constructed efficient aggregate and verifiable encrypted signatures without random oracles. From the above references, we can easily know that multiple linear pairing has been an important tool for constructing encryption and signature algorithms, and the security of the signcryption schemes is guaranteed under the multilinear Diffie-Hellman assumption, but there is almost no secret sharing scheme based on multiple linear pairing presently.

Consequently, by using multiple linear pairing, in this paper, we propose a non-interactive, simple and effective PVSS scheme, whose security is based on multilinear Diffie-Hellman assumption. In our scheme, we assume that the secret is a multiple linear paring, by using multiple linear property of multilinear map and the batch verification technique to prevent cheating at secret distribution phase and reduce computational overhead at verification phase, respectively. Moreover, anyone can identify the process of distributing and recovering secret publicly without implementing the interactive protocol such as $DLEQ(g1, h1; g2, h2)$ by Chaum and Pedersen in [5], it's an effective solution to prevent dishonest dealer and participants, thereby reducing the communication cost. Furthermore, we show that in the random oracle model and under multilinear Diffie-Hellman assumption, our proposed scheme is securely and effectively. In addition, the performance analysis shows that it is less communication overhead and more effective than the previous schemes [11, 17, 25], so it can be more applicable in those high efficiency of the communication requirements of the application scenarios.

The rest of the paper is organized as follows. We briefly describe the concept of multilinear map and the related security assumptions. At the same time, we review the model of PVSS scheme in Section 2. In Section 3, we present our new publicly verifiable secret sharing based on multiple linear pairing. And then in Section 4, we make the scheme analysis, which focus on the proof of the correctness and security, as well as the performance comparison. In Section 5, the application of our PVSS scheme in electronic voting is briefly presented. We introduce a conclusion and our next work in Section 6.

2 Preliminaries

In this section, we briefly describe the definition of multilinear map, the related security assumptions, and recall the publicly verifiable secret sharing (PVSS) scheme.

2.1 Multilinear Maps

Boneh and Silverberg (BS) [4] first proposed the concept of multilinear map and described many cryptographic applications in 2003. The definition of BS is that: Let G_1 and G_2 be two groups which have the same prime order q. In particular, G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group. A map $e : G_1^n \to G_2$ is an n-multilinear map if it satisfies the following three properties:

1) Multilinear: For all $g_1, g_2, \cdots, g_n \in G_1$ and $a_1, a_2, \cdots, a_n \in Z_q^*$, we have $e_n(a_1 g_1, a_2 g_2, \cdots, a_n g_n) = e_n(g_1, g_2, \cdots, g_n)^{a_1 a_2 \cdots a_n}$;

2) Non-degenerate: If $g \in G_1$ is a generator of G_1, then $e_n(g, g, \cdots, g)$ is a generator of G_2;

3) Computable: For all $g_1, g_2, \cdots, g_n \in G_1$, there is an efficient algorithm to compute $e_n(g_1, g_2, \cdots, g_n)$.

2.2 Security Problems and Assumptions

Computational Diffie-Hellman (CDH) problem: Given $g, ag, bg \in G_1$ for some $a, b \in Z_q^*$, it is difficult to compute $abg \in G_1$.

Discrete logarithm (DL) problem: Given $g, ag \in G_1$, it is hard to compute $a \in Z_q^*$.

Multilinear discrete logarithm (MDL) problem: Let G be a finite cyclic group with prime order q, for all $k > 1, 1 \le i \le k$ and $g_i \in G$, given (i, g_i, ag_i) for some $a \in Z_q^*$, it is hard to compute a.

n-Multilinear computational Diffie-Hellman (n-MDH) problem: Given $g, a_1 g, a_2 g, ..., a_n g \in G_1$ for some random selective $a_1, a_2, \cdots, a_n \in Z_p$, where g is a generator of group G_1, it is hard to compute $e_n(g, g, \cdots, g)^{a_1 a_2 \cdots a_n} \in G_2$.

MDH assumption: No PPT algorithm can solve the MDH problem with a non-negligible advantage.

2.3 Model of PVSS

In this section, the model of (t, n) threshold publicly verifiable secret sharing (PVSS) scheme is presented. Let t and n be two positive integers such that $1 \leq t \leq n$. Let $U_1, ..., U_n$ denote the participants, and D denotes the dealer. An access structure can be a (t, n) threshold scheme for $1 \leq t \leq n$, it means that any subset of t or more participants is able to reconstruct the secret, while the subset of at most t-1 participants cannot recover the secret and has no information about it. The system of a PVSS scheme consists of three phases are described below.

1) **Initialization phase:** On input the number n of participants, a threshold t, it outputs all public parameters as well as participants' private keys and the corresponding public keys as part of the system parameters.

2) **Distribution phase:** On input a secret s, the distribution phase consists of two steps as follows.

   a. **Share distribution:** The dealer D distributes a secret s among n participants, the dealer uses the participants' private keys and public parameters to encrypt secret and then publishes some specific value Y_i(the shares are embedded into these specific values Y_i) to the participants U_i, where for $i = 1, 2, \cdots, n$.

   b. **Public verification:** This step can be executed by a third party and determines whether the distributed shares are valid. Anyone not just the participants can verify these specific values Y_i by checking some equations. If all the checking equations hold, then these specific values Y_i are believed to be correctly published by the dealer, and the shares included in Y_i are valid. Once the equations do not hold, we say that the dealer fails to distribute a secret, and then break the scheme.

3) **Reconstruction phase:** The reconstruction phase contains decryption of the shares and reconstruction of the secret:

   a. **Decryption of the shares:** Each participant uses his/her own private key to obtain the corresponding share s_i from the specific value Y_i, respectively.

   b. **Reconstruction of the secret:** When the qualified participants offered at least t correct shares s_i, then the secret s can be recovered from these shares s_i by threshold technique such as Lagrange interpolation.

3 Proposed PVSS Scheme

In this section, we present our non-interactive and effective PVSS scheme based on multiple linear pairing.

First, the key generation center (KGC) generates m public parameters $P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)}$, $m \in {}_R Z_q^*$. We assume that the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$ will be distributed by the dealer D among n participants, where $a \in Z_q^*$. Let $U = \{U_1, U_2, \cdots, U_n\}$ be a set of n qualified participants. The PVSS scheme consists of three phases: Initialization phase, Distribute phase and Reconstruct phase.

1) **Initialization phase**
   Let G_1 and G_2 be two groups, separately denote additive cyclic group and multiplicative cyclic group which have the same prime order q. Assuming that there exists a multilinear map $e : G_1^n \to G_2$ among G_1 and G_2. The independently generators P, Q of groups G_1 and G_2 are selected using appropriate public procedure. Each participant U_i chooses a private key $d_i \in Z_q^*$ and compute the corresponding public key $P_i = d_i P_{pub}^{(i)}$ for $i = 1, 2, ..., n$.

2) **Distribute phase**
   The distribution phase consists of two steps as following:

   a. **Distribution of the shares:** The dealer D wishes to distribute a secret among n participants. The dealer D first chooses a random polynomial $f(x) = \sum_{j=0}^{t-1} a_j x^j$ of degree at most $t-1$ with coefficients in Z_q. Here $f(0) = a_0 = a$. And then the dealer keeps this polynomial secretly but computes and publishes the following values: the related commitments $C_j = a_j \cdot P$, for $j = 0, 1, ..., t-1$, $X_i = f(i) \cdot P$ and $\gamma_i = f(i) \cdot P_{pub}^{(i)}$. The dealer also publishes the encrypted shares $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}..., P_{pub}^{(m)})^{f(i)}$ for $i = 1, 2, ..., n$.
   Each X_i can be constructed by all public values C_j as follows:

$$\begin{aligned} X_i &= f(i) \cdot P \\ &= \sum_{j=0}^{t-1} a_j \cdot (i^j) \cdot P \\ &= \sum_{j=0}^{t-1} (i^j) \cdot a_j \cdot P \\ &= \sum_{j=0}^{t-1} (i^j) \cdot C_j \end{aligned}$$

   b. **Verification of shares:** Anyone first can recover $X_i = \sum_{j=0}^{t-1} (i^j) \cdot C_j$ from the value C_j and then checks equation (1) by public values C_j and X_i, γ_i, for $j = 0, 1, ..., t-1$, $i = 1, 2, ..., n$. Equation (1):

$$\begin{aligned} &e_m(\gamma_1, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, X_j) \\ &= e_{m+1}(\gamma_1, \gamma_2, ..., \gamma_{m-1}, P) \end{aligned} \quad (1)$$

If the Equation (1) holds, then the verifier believes that these specific values Y_i correctly published by the dealer D and the verifier can confirm that each Y_i holds for $i = 1, 2, ..., n$. The proof is as follows:

$$\begin{aligned} Y_i &= e_m(P_i, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{f(i)} \\ &= e_m(d_i \cdot P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{f(i)} \\ &= e_m(P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, ..., P_{pub}^{(m)})^{d_i \cdot f(i)} \\ &= e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{d_i \cdot f(i)}. \end{aligned}$$

3) **Reconstruct phase**
This phase is divided into decryption of the shares and the reconstruction of the secret:

a. **Decryption of the shares:** Each participant U_i uses his/her own private key d_i to compute the corresponding share $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)}$ by computing the following equation:

$$\begin{aligned} {Y_i}^{d_i^{-1}} &= e_m(P_i, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{f(i) \cdot d_i^{-1}} \\ &= e_m(d_i \cdot P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{f(i) \cdot d_i^{-1}} \\ &= e_m(P_{pub}^{(i)}, P_{pub}^{(1)}, ..., P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{f(i)} \\ &= e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)} \\ &= S_i. \end{aligned}$$

b. **Reconstruct of the secret:**
Any t shareholders U_i with the correct shares S_i can reconstruct the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$, for $i = 1, 2, ..., t$. The secret S is obtained by Lagrange interpolation as Equation (2):

$$S = \prod_{i=1}^{t} S_i^{\lambda_i} = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a \quad (2)$$

Where $\lambda_i = \prod_{j \neq i} \frac{i}{j-i}$ is Lagrange coefficient.

4 Scheme Analysis

This section introduced the proof of the correctness and security of the proposed scheme, and we make performance analysis mainly in the computation and communication aspects.

4.1 Correctness Analysis

Lemma 1. *First, we verify the equation* $e_m(\gamma_1, ..., \gamma_{j-1}, P_{pub}, \gamma_{j+1}, ..., \gamma_{m-1}, X_j) = e_m(\gamma_1, \gamma_2, ..., \gamma_{m-1}, P)$ *(1).*

Proof. From the public values $X_i = f(i) \cdot P$, $\gamma_i = f(i) \cdot P_{pub}^{(i)}$. we can gain that

$$\begin{aligned} &e_m(\gamma_1, \gamma_2, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, X_j) \\ &= e_m(\gamma_1, \gamma_2, ..., \gamma_{j-1}, P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, f(j) \cdot P) \\ &= e_m(\gamma_1, \gamma_2, ..., \gamma_{j-1}, f(j) \cdot P_{pub}^{(j)}, \gamma_{j+1}, ..., \gamma_{m-1}, P) \\ &= e_m(\gamma_1, \gamma_2, ..., \gamma_{j-1}, \gamma_j, \gamma_{j+1}, ..., \gamma_{m-1}, P) \\ &= e_m(\gamma_1, \gamma_2, ..., \gamma_{m-1}, P). \end{aligned}$$

Hence, Equation (1) holds, the shares distributed by the dealer are valid. □

Lemma 2. *And then verify that the method of reconstructing the secret is correct. In other words, it is need to verify equation* $S = \prod_{i=1}^{t} S_i^{\lambda_i} = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$.

Proof. From the known share value $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)}$, which is computed from private key d_i and specific public value Y_i, we can get that

$$\begin{aligned} \prod_{i=1}^{t} S_i^{\lambda_i} &= \prod_{i=1}^{t} (e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)})^{\lambda_i} \\ &= e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{\sum_{i=1}^{t} f(i) \cdot \lambda_i} \\ &= e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)} \\ &= e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a \\ &= S \end{aligned}$$

The Equation (2) holds, so the method of secret reconstruction is correct. □

4.2 Security Analysis

In this section, we present security analysis of our proposed scheme under the multilinear Diffie-Hellman (MDH) assumption.

We first consider the security of the shares $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$. Given the public values $P_{pub}^{(i)}, P_i, X_i$ and Y_i for $i = 1, 2, ..., n$, we observe that the difficulty of computing the share S_i is equivalent to solve the multilinear Diffie-Hellman (MDH) problem as described in Section 2. Consequently, we have the following lemma.

Lemma 3. *The encryption of shares is security in the proposed PVSS scheme if and only if the MDH assumption holds.*

Proof. ⇐ By contradiction proof. Assuming that the MDH assumption holds but the encryption of shares is not security. Since the method of share encryption does not hold, then there exists an Algorithm A can compute the shares $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$ with a

non-negligible probability ε for the given public value $P_{pub}^{(i)}, P_i, X_i$ and Y_i. Then we want to prove that an attacker can solve the MDH problem with the same probability using the Algorithm A.

The MDH problem is that given $a_1P, a_2P, ..., a_mP$ for some $a_1, a_2, \cdots, a_m \in Z_q^*$, it is hard to compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$. Hence, we try to compute the value $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ using A in the following. The attacker chooses random elements $a_1, a_2, \cdots, a_m, b \in Z_q^*$ and $a_1^{'}, a_2^{'}, \cdots, a_m^{'}, b' \in Z_q^*$. For the given values $Q_1 = a_1P, Q_2 = a_2P, ..., Q_m = a_mP, Q = bP$, the attacker first computes and feeds the values $P_{pub}^{(i)} = a_i^{'} \cdot Q_i, P_i = a_i \cdot P_{pub}^{(i)}$, $X_i = b' \cdot Q$ and $Y_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{a_i \cdot bb'}$ to A, where $i = 1, 2, ..., n$. Since the input of A is uniformly distributed and $X_i = b' \cdot Q = b' \cdot bP = f(i)P$ is known, we obtain that $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)} = e_m(a_1a_1^{'} \cdot P, a_2a_2^{'} \cdot P, ..., a_ma_m^{'} \cdot P)^{f(i)} = e_m(P, P, ..., P)^{a_1a_1^{'} \cdot a_2a_2^{'} \cdot a_ma_m^{'} \cdot f(i)} = e_m(P, P, ..., P)^{a_1a_1^{'} \cdot a_2a_2^{'} \cdot a_ma_m^{'} \cdot bb'}$ with the same non-negligible probability ε. By taking $(S_i)^{1/(a_1^{'}a_2^{'}\cdots a_m^{'}bb')} = e_m(P, P, ..., P)^{a_1a_2\cdots a_m}$, we know that the attacker is able to compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ with the probability ε. It is a contradiction to the above MDH assumption.

It shows that the MDH assumption holds, then the encryption of shares is secure.

$\Rightarrow$ By contradiction proof. Assuming that the encryption of shares is secure but the MDH assumption does not hold. Because the MDH assumption does not hold, then there exists an algorithm B can compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ with a non-negligible probability ε for m random elements $a_1P, a_2P, ..., a_mP \in G_1$, where $a_1, a_2, \cdots, a_m \in Z_q^*$. The attacker chooses random elements $\beta_1, \beta_2, \cdots, \beta_m, b \in Z_q^*$ and $\beta_1^{'}, \beta_2^{'}, \cdots, \beta_m^{'}, b' \in Z_q^*$. When feeding $Q = bP, X_i = b'Q$ to B, the attacker computes and inputs $Q_1^{'} = \beta_1 \cdot P, Q_2^{'} = \beta_2 \cdot P, ..., Q_m^{'} = \beta_m \cdot P$, $P_{pub}^{(i)} = \beta_i^{'}P$ for $i = 1, 2, ..., n$. Then the share S_i must satisfy that $S_i = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{f(i)} = e_m(Q_1^{'}, Q_2^{'}, \cdots, Q_m^{'}) = e_m(\beta_1P, \beta_2P..., \beta_mP)$.

Since the input of B is uniformly distributed, we can compute $X_i = b'Q = b'bP = f(i)P$ with the same probability ε because of $Q = bP, X_i = b'Q$. Therefore, we can obtain that $e_m(\beta_1^{'}P, \beta_2^{'}P, ..., \beta_m^{'}P)^{f(i)} = e_m(\beta_1P, \beta_2P..., \beta_mP)$. Which produces

$$e_m(P, P..., P)^{\beta_1^{'}\beta_2^{'}\cdots\beta_m^{'}bb'} = e_m(P, P..., P)^{\beta_1\beta_2\cdots\beta_m}.$$

Due to the MDH assumption does not hold, So the algorithm B can compute $e_m(P, P..., P)^{\beta_1\beta_2\cdots\beta_m}$ with the same non-negligible probability ε, and then the share S_i can be computed by algorithm B. Hence, the encryption of shares is not secure.

It shows that the encryption of shares is secure, the MDH assumption must hold. □

Lemma 4. *If only t-1 participants can work together to recover the secret in the proposed scheme, then the Multilinear Diffie-Hellman (MDH) problem can be solved.*

Proof. At first, we recall that the MDH problem is to compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ for given $P, a_1P, a_2P, ..., a_nP \in G_1$ for some random choices $a_1, a_2, \cdots, a_n \in Z_p$. As in Section 2, solving the MDH problem is to compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ with the non-negligible probability ε.

Without loss of generality, we assume that t-1 participants $U_1, U_2, ..., U_{t-1}$ are able to pool their valid shares and recover the secret.

Now we need to prove that adversary Λ can compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ by using t-1 participants as oracle. In the following, we will set up the system to simulate PVSS for adversary Λ such that this system enables the adversary Λ to compute $e_m(P, P, \cdots, P)^{a_1a_2\cdots a_m}$ when t-1 participants are seen as oracle. The Setup system consists of six steps as follows:

1) Adversary Λ sets $P_{pub}^{(i)} = a_iP, C_0 = bP(= f(0)P)$ for $i = 1, 2, ..., n$, where $a_i \in Z_q^*, b \in Z_q^*$.

2) Taking t-1 values: The values $f(1), f(2), ..., f(t-1)$ are chosen at random from Z_q^*, and previous fixed $f(0)$ such that a polynomial $f(x)$ can be fixed.

3) Adversary Λ compute forward t-1 values of X_i and Y_i as follows: $X_i = f(i)P$, $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)} ..., P_{pub}^{(m)})^{f(i)}$, $i = 1, 2, ..., t-1$.

4) $f(0)$ is hiding fixed, so Λ is not able to compute the following values: $f(t), f(t+1), \cdots, f(n)$. However, we can use X_i for $i = 1, 2, \cdots, t-1$ to obtain C_j by solving t-1 simultaneous equations $X_i = \sum_{j=0}^{t-1} (i^j) \cdot C_j$ for $j = 1, 2, \cdots, t-1$. When we have computed these values C_j, we can obtain X_i for $i = t, t-1, ..., n$ by Lagrange interpolation formula.

5) First compute $C_j (i = 1, ..., t-1)$. Since $f(x) = \sum_{i=1}^{t-1} a_i \cdot x^i$, then there is the following linear system of equations:

$$\begin{cases} f(0) = a_0 \\ f(1) = a_0 + a_1 \cdot 1 + \cdots + a_{t-1} \cdot 1^{t-1} \\ \vdots \\ f(t-1) = a_0 + a_1 \cdot (t-1)^1 + \cdots + a_{t-1} \cdot (t-1)^{t-1} \end{cases}$$

In this linear system of equations, adversary Λ knows the values of $f(1), f(2), \cdots, f(t-1)$, while $f(0)$ is unknown, so it is unable to compute the coefficient a_i of the polynomial $f(x)$. However, adversary Λ can compute values of C_j by the linear system of equations and public values C_0, X_j for $i = 0, 1, \cdots, t-1, j = 1, 2, ..., t-1$.

6) Now, adversary Λ computes the public keys P_i of participants U_i as $P_i = v_i \cdot P_{pub}^{(i)}$ for $i = 0, 1, ..., n$, where $v_i \in Z_q^*$. In particular, we set $Y_i = e_m(X_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(m)})^{a_i \cdot v_i}$ such that $Y_i = e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)} ..., P_{pub}^{(m)})^{f(i)}$, as required.

Now, the complete view for the system is defined. Which is consistent with the private view of participants $U_1, U_2, ..., U_{t-1}$, and the view comes from the right distribution. Supposing that they are able to obtain the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)}$. Since $P_{pub}^{(i)} = a_i P$ and $f(0) = b$ for $i = 0, 1, \cdots, n$, we can compute $e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(0)} = e_m(P, P, \cdots, P)^{a_1 a_2 \cdots a_m b}$. It contradicts the MDH assumption. □

So far we have ignored the proofs that are required at several points in the protocol. However, in the random oracle model, these proofs can easily be simulated. By the above two lemmas, we can draw the following theorem.

Theorem 1. *Under the MDH assumption, the proposed scheme is a secure PVSS scheme in the random oracle model. That is, (1) only qualified participants can compute the valid shares; (2) any subset of t-1 participants is unable to recover the secret. (3) The proposed PVSS scheme must provide publicly verifiable property.*

1) From Lemma 3 and the scheme's construction method in Section 3, we know that $Y_i = S_i^{d_i}$, then any attacker is unable to compute the corresponding shares S_i from these specific values Y_i because of the hardness of the MDH and discrete logarithm.

2) By Lemma 4, any t participants with shares S_i can obtain the secret $S = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^a$ by Lagrange interpolation method. And any subset of t-1 or less participants is unable to recover the secret unless the MDH problem is solved.

3) From Section 3, it is easy to know that anyone not just the participants can verify each Y_i whether it is equal to $e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{d_i f(i)}$ with the dealer's secret $f(i)$ for $i = 0, 1, \cdots, n$. In this section, we also have verified that each qualified participant U_i can use his/her private key d_i to compute the share $S_i = Y_i^{d_i^{-1}} = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, ..., P_{pub}^{(m)})^{f(i)}$. Each S_i also contains the factor $f(i)$. So the proposed scheme must provide publicly verifiable property.

4.3 Performance Analysis

In this subsection, we mainly analyze the computation overhead and communication overhead. The performance analysis shows that our scheme is effective when comparing with previous schemes. For convenience to evaluate the computational cost, we define the following notations:

T_{e_m}**:** The time of executing a multiple linear pairing operation $e_m : G_1^m \to G_2$.

TG_{mul}**:** The time of executing a scalar multiplication operation of points in G_1.

$T_{\exp}$**:** The time of executing a modular exponent operation of points in Z_q.

T_{mul}**:** The time of executing a modular multiplication operation of points in Z_q.

T_{Lag}**:** The time of using the Lagrange interpolating method to construct the secret.

T_{pol}**:** The time of computing the polynomial value $f(x) = \sum_{i=0}^{t-1} a_i x^i$ in Z_q.

1) **From the computation aspect.** As we all know, the most time consuming is power modular operation in the scheme based on Discrete Logarithm. The most time consuming is a modular multiplication operation of points in the scheme based on ECDLP. While the most time consuming mainly contains T_{e_m}, TG_{mul} in the scheme based on multiple linear pairing.

Hence, we only consider these time-consuming operations T_{e_m}, TG_{mul} and T_{exp} in the performance analysis of the proposed PVSS scheme. In our scheme, there is no need for the dealer to compute the corresponding shares for the participants, compared with the references [23] and [14], our scheme solves the overhead at the secret distribution phase. Especially in verification phase of the shares, we use the tool of multiple linear paring and the batch verification technique to reduce the computational overhead. In Table 1, we list the performance comparison, which are concentrated on the publicly verifiability and computation cost of all phases in secret sharing schemes.

From Table 1, we know that the computation overhead of our scheme is lower in share verification phase, and the main operation cost is a linear relationship with the number of participants. In addition, some calculations can be done preprocessing in secret distribution phase, which can greatly improve the efficiency of secret distribution.

2) **From the communication aspect.** Since there is no need for our scheme to implement interactive protocol to prevent malicious players, which greatly saves the communication overhead. The communication complexity of our scheme is lower than PVSS in [19]. The communication of the proposed scheme mainly reflects in secret distribution phase and reconstruction phase. Namely the process of the dealer distributes the secret and publishes the public information at secret sharing phase, as well as the overhead of t shareholders pool shares to the secret restorer. Other phases do not need interaction between participants. Consequently, the total communication cost of our scheme is $4nq + tq$, which is almost the

Table 1: Performance comparison

Authors' schemes	Publicly verifiable	The computation cost of all phases		
		Distribution phase	Verification phase	Construction phase
Tian et al. [22]	No	$t(n+1)TG_e + nT_{pol}$ $+2tTG_{mul}$	$TG_e + TG_{mul} + tT_{\exp}$	$T_{Lag} + tT_{mul}$
Wu et al. [25]	Yes	$TG_e + (4n+t)TG_{mul}$ $+nT_{\exp} + nT_{pol}$	$(n+3)TG_e + n(t+1)TG_{mul}$ $+nT_{\exp} + ntT_{pol}$	$T_{Lag} + tT_{\exp}$
Tian et al. [24]	Yes	$nTG_e + 2nTG_{mul}$ $+nT_{mul} + nT_{pol}$	$nTG_e + nTG_{mul}$ $+nT_{\exp} + ntT_{pol}$	$T_{Lag} + tT_{mul}$
Our PVSS	Yes	$(2n+1)TG_e + 3nTG_{mul}$ $+nT_{\exp} + nT_{pol}$	$2nTG_e + nT_{mul}$	$T_{Lag} + tT_{\exp}$

same with reference [25]. Moreover, using multiple linear paring and the technique of batch verification, our communication overhead has great advantage at the share verification phase. So the proposed scheme is less communication overhead and more effective.

5 Discussion

In this section, the application of our publicly verifiable secret sharing scheme is presented in electronic voting. By using our PVSS scheme as a basic tool, we get a simple and efficient voting scheme. At last, we analyze the advantages of this electronic voting scheme.

From the model for universally verifiable elections as introduced by Hwang et al. [15], it is easily to know that all of the players will post their messages in electronic voting schemes. We assume that the players are composed by a set of tallying authorities (tallliers) $T_1, \cdots, T_n$, which act as the participants in our PVSS scheme, a set of Voters $V_1, \cdots, V_l$, and each of them acts as a dealer in our PVSS scheme, as well as a set of passive observers. These sets need not be disjoint, each player may be both a voter and a tallier. Assuming that each tallier T_i has registered a public key $P_i = a_i P_{pub}^{(i)}$ for the randomly selected private key $a_i \in {}_R Z_q$, where $i = 1, 2, \cdots, n$.

The designed electronic voting scheme consists of two phases: Ballot casting and Tallying.

1) **Ballot casting.** A voter V casts a vote $v \in \{0, 1\}$ by running the distribution protocol for our PVSS scheme from Section 3, using a random secret value $a \in {}_R Z_q$, the voter can compute the value $U = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{a+v}$. In addition, the voter constructs a proof $PROOF_U$ showing that indeed $v \in \{0, 1\}$ without revealing any information on v. $PROOF_U$ refers to the commitment value of $C_0 = a_0 P = aP$ which is published as part of the PVSS distribution protocol. And then each voter proves that: $e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots P_{pub}^{(m)}) = e_m(P_{pub}^{(1)}, P_{pub}^{(2)}, \cdots, P_{pub}^{(m)})^{a_i+v}$.

Due to the publicly verifiability of the proposed PVSS scheme and the known value of $PROOF_U$, the ballots can be checked by using the above equation by the bulletin board when the voters submit their ballots. What's more, the ballot for voter V consists of the output values U and $PROOF_U$ of the PVSS distribution protocol.

2) **Tallying.** Supposing that voters V_j have all cast valid ballots, where $j = 1, \cdots, k$ and $k \leq l$. The tallying protocol uses the reconstruction protocol of our PVSS scheme. We first accumulate all the respective encrypted shares, that is, we compute the values Y_i^*, where

$$\begin{aligned} Y_i^* &= \prod_{j=1}^{k} Y_{ij} \\ &= e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \\ &\qquad \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} f_j(i)}. \end{aligned}$$

And then each tallier T_i applies the reconstruction protocol to the value Y_i^*, which will produce

$$\begin{aligned} & e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} f_j(0)} \\ = \; & e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{a_j} \end{aligned}$$

Next, by combining with the equation

$$\begin{aligned} & \prod_{j=1}^{k} U_j \\ = \; & e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \\ & \qquad \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} a_j + v_j}. \end{aligned}$$

We obtain

$$e_m(P_i, P_{pub}^{(1)}, \cdots, P_{pub}^{(i-1)}, P_{pub}^{(i+1)}, \cdots, P_{pub}^{(m)})^{\sum_{j=1}^{k} v_j},$$

from which the tally $T = \sum_{j=1}^{k} v_j, 0 \leq T \leq k$ can be computed efficiently.

The advantages of this electronic protocol:

1) In the ballot casting phase, the voters' ballots contain the votes in encrypted form and the voters need not be anonymous in this protocol. In tallying phase, the talliers use their private keys to collectively compute the final tally corresponding with the accumulation of all the valid ballots.

2) The above electronic voting scheme achieves the same level of security with regard to publicly verifiability, privacy, and robustness.

3) Our scheme does not require a shared-key generation protocol for a threshold decryption scheme, which avoids the interaction between the voters and the interaction among the talliers.

4) Compared with [9], which requires a private channel by public key encryption, our protocol does not need a shared-key generation protocol, so the information-theoretic privacy for the voters is not lost.

Analysis results show that our PVSS scheme can be used in elections for computational privacy without needing a private channel.

6 Conclusion

In this paper, we proposed a non-interactive, simple and effective publicly verifiable secret sharing based on multiple linear pairing. In our PVSS scheme, not just the participants, anyone is able to verify whether the shares distributed by the dealer are correctly at the secret distribution phase and whether each participant releases valid shares at the reconstruction phase. We use multiple linear property of multilinear map and the batch verification technique to reduce the computational overhead at verification phase. The computation cost and communication overhead are lower than the previous PVSS schemes which are based on bilinear paring or discrete logarithm. In addition, under the multilinear Diffie-Hellman assumption, we have shown our PVSS scheme is security in the random oracle model. In the discussion section, we present the application of our PVSS scheme in electronic voting and analyze the advantages of this protocol. Our next work is to apply the proposed PVSS scheme in secure multi-party computation and other practical protocols.

Acknowledgments

This study was supported by The National Natural Science Foundation of China under Grant (No. 61363068, 61262073), the China Postdoctoral Science Foundation under Grant (No. 2013M530705), the National Natural Science Foundation of Guizhou under Grant (No. 20132112), the Doctor Foundation of the Guizhou University under Grant (No. 2012-024). The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] C. Bhagvati, "CRT based threshold multi secret sharing scheme," *International Journal of Network Security*, vol. 16, no. 4, pp. 249–255, 2014.

[2] G. R. Blakley et al., "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, vol. 48, pp. 313–317, 1979.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (Crypto'01)*, pp. 213–229. Springer, 2001.

[4] D. Boneh and A. Silverberg, "Applications of multilinear forms to cryptography," *Contemporary Mathematics*, vol. 324, no. 1, pp. 71–90, 2003.

[5] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," in *Advances in Cryptology (Eurocrypt'92)*, pp. 390–407, Springer, 1993.

[6] H. Y. Chien, J. K. Jan and Y. M. Tseng, "A unified approach to secret sharing schemes with low distribution cost," *Journal of the Chinese Institute of Engineers*, vol. 25, no. 6, pp. 723–733, 2002.

[7] H. Y. Chien, J. A. N. Jinn-Ke and Y. M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 12, pp. 2762–2765, 2000.

[8] J. S. Coron, T. Lepoint and M. Tibouchi, "Practical multilinear maps over the integers," in *Advances in Cryptology (Crypto'13)*, pp. 476–493, Springer, 2013.

[9] R. Cramer, M. Franklin, B. Schoenmakers and M. Yung, "Multi-authority secret-ballot elections with linear work," in *Advances in Cryptology (Eurocrypt'96)*, pp. 72–83, Springer, 1996.

[10] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[11] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th IEEE Annual Symposium on Foundations of Computer Science*, pp. 427–438, 1987.

[12] S. Garg, C. Gentry and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Eurocrypt*, vol. 7881, pp. 1–17, Springer, 2013.

[13] S. Garg, C. Gentry, S. Halevi, A. Sahai and B. Waters, "Attribute-based encryption for circuits from multilinear maps," in *Advances in Cryptology (Crypto'13)*, pp. 479–499, Springer, 2013.

[14] L. Harn, "Efficient sharing (broadcasting) of multiple secrets," *IEE Proceedings-Computers and Digital Techniques*, vol. 142, no. 3, pp. 237–240, 1995.

[15] C. T. Li, M. S. Hwang and Y. C. Lai, "A verifiable electronic voting scheme over the internet," in *IEEE Sixth International Conference on Information Technology: New Generations* , pp. 449–454, 2009.

[16] A. J. Menezes, T. Okamoto, S. Vanstone, et al., "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transactions on Information Theory* , vol. 39, no. 5, pp. 1639–1646, 1993.

[17] 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.

[18] M. Rückert and D. Schröder, "Aggregate and verifiably encrypted signatures from multilinear maps without random oracles," in *Advances in Information Security and Assurance*, pp. 750–759, Springer, 2009.

[19] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology (Crypto'99)*, pp. 148–164, Springer, 1999.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[21] M. Stadler, "Publicly verifiable secret sharing," in *Advances in Cryptology (Eurocrypt'96)*, pp. 190–199, Springer, 1996.

[22] Y. L. Tian, J. F. Ma, C. G. Peng and X. Chen, "Information-theoretic secure verifiable secret sharing scheme on elliptic curve group," *Journal on Communications*, vol. 12, pp. 014, 2011.

[23] Y. Tian and C. Peng, "Verifiable secret sharing scheme and applications based on bilinear pairing," *Computer Engineering*, vol. 35, no. 10, pp. 158–161, 2009.

[24] Y. Tian, C. Peng and J. Ma, "Publicly verifiable secret sharing schemes using bilinear pairings," *International Journal Network Security*, vol. 14, no. 3, pp. 142–148, 2012.

[25] T. Y. Wu and Y. M. Tseng, "A pairing-based publicly verifiable secret sharing scheme," *Journal of Systems Science and Complexity*, vol. 24, no. 1, pp. 186–194, 2011.

[26] J. Yu, F. Kong and R. Hao, "Publicly verifiable secret sharing with enrollment ability," in *IEEE Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, vol. 3, pp. 194–199, 2007.

Qiao Peng born in 1991, M.S candidate. Her research interests include information security and cryptography.

Youliang Tian born in 1982, the corresponding author, PH.D, associate professor and M.Sc supervisor. Now he serves as deputy director of the institute of information and data security of guizhou university. And his research interests focus on game theory, information security and cryptography.

Reviewers (Volume 18, 2016)

Geetha Mary A
Fathi Abd El-Samie
Abdeldime Mohamed Salih
Abdelgader
Mohd Faizal Abdollah
Shafi'i Muhammad
Abdulhamid
Ashwini B Abhale
Qasem Abu Al-Haija
Udaya Kumar Addanki
Syed Hasan Adil
Abdolkarim Afroozeh
Sarita Agrawal
Muhammad Najmi Ahmad
Zabidi
Mohammad Reza Ahmadi
Irfan Ahmed
Asimi Ahmed
Ganesh V Aithal
Shahid Alam
Monjur M Alam
Sara Ali
K.A.Nusrath Ali
Ali Mohamed Allam
Ruhul Amin
Rengarajan Amirtharajan
Benjamin Arazi
Razi Arshad
Dawar Asfandyar
Heba Kamal Aslan
Felix Au-Yeung
Amit K Awasthi
Karthikeyan B
Parameshachari B D
Ayman Bahaa
Saeed Bahmanabadi
Kavitha Balu
Pijush Barthakur
Eihab B Bashier
Sunny Behal
Mohammad Beheshti
Atashgah
Massimo Bernaschi
Jaydeb Bhaumik
Krishna Bhowal
Monowar H. Bhuyan
Zulkhar Nain Bin Badruz
C. Shoba Bindhu
Santosh Biswas
Andrew Blyth
Mitko Bogdanoski
Tianjie Cao
Zhenfu Cao
Zhengjun Cao
Rui Costa Cardoso
Veerendra Kumar Ch
Chi-Shiang Chan
Chin-Chen Chang
Ayantika Chatterjee
Ali M Chehab
Zhen Hua Chen
Jan Min Chen
Yi-Hui Chen
Zhenhua Chen
Tzung-Her Chen
Hu Chengyu
Abbas Cheraghi
Rachid Cherkaoui
Hongmei Chi
Hung-Yu Chien
Mohamedsinnaiya Chithikraja
Hassan Chizari
Kim-Kwang Raymond Choo
Deepak Chou
Jue-Sam Chou
Aileen P Chris
Christopher P. Collins
Nicolae Constantinescu
Manivannan D
Anil Kumar Dahiya
Hisham Mohammed Dahshan
Jawad Ahmad Dar
Ashok Kumar Das
Lunzhi Deng
Subhasish Dhal
Alberto Pe Dominguez
Lihua Dong
Nishant Doshi
Yunqi Dou
Yassine Douga
Ahmed Drissi
Qi Duan
Munivel E
Nawal El-Fishawy
Bou-Harb Elias
Claude Fachkha
Mohammad Sabzinejad
Farash
Yue Feng
Xingbing Fu
Nisarg Gajanan Gandhewar
Rakesh C Gangwar
Amzari Ghazali
Debasis Giri
Faiq Gmira
M C Gorantla
Xiaozhuo Gu
Avinash K Gulve
T Gunasekhar
Rui Guo
C P Gupta
Shashi Gurung
Sandeep Raj Gurung
Arash Habibi Lashkari
Nashrul Hakiem
Tanmoy Halder
Yasir Hamid
Disha Handa
Charifa Hanin
Lein Harn
Syed Hamid Hasan
Seyed Hashemi
Ali Hassan
Amir Hassani Karbasi
Thaier Hayajneh
Luis Hernandez
Carlos Hernandez-Castro
Gwoboa Horng
Osama Hosam Eldeen
Defa Hu
Xiong Hu
Yen-Hung Hu
Peng Hu
Xuexian Hu
Huajun Huang
Chin-Tser Huang
Phoenix Huang
Osamah Ibrahiem
Maged Hamada Ibrahim
Rajesh Ingle
Sk Hafizul Islam

Grasha Jacob
Biswapati Jana
Tsai Jia Lun
Lin Zhi Jiang
Shaoquan Jiang
Rui Jiang
Qi Jiang
Zeng Jianping
Wang Jie
Zhengping Jin
Ashish Joshi
Adri Jovin
Peyman Kabiri
Ajay Kakkar
Yoshito Kanamori
Akram Kargar Raeespour
Rasool Kazemi
Peiman Keshavarzian
Sumit A Khandelwal
Walid Ibrahim Khedr
Behbod Kheradmand
Dong Seong Kim
Jung-Tae Kim
Brian King
Basappa Bharamappa Kodada
Anjan Krishnamurthy
Pramote Kuacharoen
Sajja Ratan Kumar
Beesetti Kiran Kumar
Yogesh Kumar
K S Anil Kumar
Naresh N Kumar
Deepak Kumar
Manoj Kumar
Saru Kumari
Virendra Singh Kushwah
Mohamad I Ladan
Shahram Latifi
Phu Le
Cheng-Chi Lee
Jung-San Lee
Keying Li
Qi Li
Jiping Li
Chun-Ta Li
Qinyi Li
Jiguo Li
Tian You Liang
Kriangkrai Limthong
Chia-Chen Lin
Chih-Yang Lin
Yang-Bin Lin
Jeng-Ping Lin
Tzu-Chun Lin
Ximeng Liu
Liang Liu
Gao Liu
Guanfeng Liu
Zhe-Ming Lu
Rongxing Lu
Jun Luo
Ming Luo
Lintao Lv
Hemalatha M
Sanikommu Madhavi
Sagar Bhaskar Mahajan
Tanmoy Maitra
Yassine Maleh
Hafiz Malik
Arun Malik
Palvinder Singh Mann
Qurban A Memon
Bo Meng
Yang Ming
Amit Mishra
Adhani Mj
M A Mohamed
Ferrag Mohamed Amine
Mohammed Ramadan
Mohammed
Guillermo Morales-Luna
Hamdy M. Mousa
Arif - Muntasa
Zulkiflee Muslim
Amitava Nag
Preeti Nagrath
Loris Nanni
Syed Naqvi
Kanagaraj Narayanasamy
Anand Nayyar
Sarmistha Neogy
Amin Nezarat
Siaw-Lynn Ng
Xuyun Nie
Krishnamur G Ningappa
Mahda Noura
Tanvir Orakzai
Jidesh P
Sahadeo - Padhye
Nasrollah Pakniat
Francesco Palmieri
Mrutyunjaya Panda
Subhash S Parimalla
Arun Raj Kumar Parthiban
Kanubhai K Patel
Kailas Ravsaheb Patil
Gerardo Pelosi
Fei Peng
Changgen Peng
Chuan Qin
Kashif Naseer Qureshi
Hani Qusa
Narasimhan Renga Raajan
Benjamin W. Ramsey
Anurag Rana
Rama Chandra Rao
Vimalathithan
Rathinasabapathy
Dhivya Ravi
Prabhudutta Ray
Jianguo Ren
Yanli Ren
Habib Rostami
B A Sabarish
Yassine Sadqi
Magdy M. Saeb
Maryam Saeed
Mounita Saha
Mahmoud Salmasizad
Deny J Sam
Sabyasachi Samanta
Arun Kumar Sangaiah
S Santhanalakshmi
Arindam Sarkar
Behnaz Saropourian
Sathish Ku Sathish Ku
Ashutosh Saxena
Neetesh Saxena
Akkapeddi Chandra Sekhar
Resmi Sekhar
Thamizh D Selvam
Irwan Sembiring
Elena Sendroiu
Seyyed Amin Seyyedi
Aamir Shahzad
Kareemulla Shaik
Anish Shandilya
Xiao Shangqin
Tarun Narayan Shankar
Udhayakumar Shanmugam

Rohith Shivashankar
Jin Shu
Jitendra Singh
Sandeep Singh
Gurpal Singh
Nicolas Sklavos
Hamdy S Soliman
Balaji P Srikaanth
Miroslav Stampar
Deris Stiawan
Weiqing Sun
Lathies Bhasker T
Nedal Mohammad Tahat
Zuowen Tan
Maryam Tanha
Ariel Soares Teles
Abebe Tesfahun
John Lane Thames
Tony Thomas
Miaomiao Tian
Xiuxia Tian
Rajesh Kumar Tiwari
Geetam Singh Tomar
Zouheir Mustapha Trabelsi
Yuan-Yu Tsai
S.C. Tsaur
Venkanna U
Mangesh Rameshrao Umak
Subba Rao Y V
Raghav V. Sampangi
Pallapa Venkataram
S.Maria Celestin Vigila
Samuel S. Wagstaff J
Osman Wahballa
Feng Wang
Yiming Wang
Ying Wang
Yong Wang
Shuozhong Wang
Libin Wang
Daxing Wang
Ding Wang
C. H. Wei
Fushan Wei
Jianghong Wei
Chenhuang Wu
Shyi-Tsong Wu
Christos Xenakis
Qi Xie
Jinbo Xiong
Lei Xu
Chunxiang Xu
Zhao Xu
Narayanan Arumugam Yadhav
Xu Yan
Zheng Yang
Li Yang
Jun Ye
Venkatramana Reddy Yeddula
Tzu-Chang Yeh
Huang Yiwang
Lin You
Milad Yousefi
Zhou Yousheng
Huifang Yu
M. Zaki
Sherali Zeadally
Shengke Zeng
Jianping Zeng
Xiaojun Zhang
Futai Zhang
Yinghui Zhang
Shun Zhang
Mingwu Zhang
Jie Xiu Zhang
Wen Zhang
Fangguo Zhang
Xingwen Zhao
Gansen Zhao
Ming Zhao
Wang Zhi-Hui
Zhiping Zhou
Frank Zhu
Ye Zhu
Hongfeng Zhu

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.jalaxy.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].

Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 7,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijns.jalaxy.com.tw or Email to ijns.publishing@gmail.com.