IJNS
International Journal of Network Security

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

Vol. 18, No. 5 (Sept. 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. 5 (Sept. 1, 2016)

Provably Secure Multi-server Privacy-Protection System Based on Chebyshev Chaotic Maps without Using Symmetric Cryptography

Hongfeng Zhu, Yifeng Zhang, and Yang Sun
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, P. R. China
(Email:zhuhongfeng1978@163.com, {1548452125, 17247613}@qq.com)
(Received Mar. 15, 2015; revised and accepted May 5 & July 13, 2015)

Abstract

Most of the privacy-protection schemes adopting chaotic maps are usually by symmetric cryptography for guaranteeing identity hiding. This will lead to a high calculated amount. So, the paper will wipe out the symmetric cryptography, and only use chaotic maps, a secure one-way hash function to construct a provable privacy-protection system (PPS) which can achieve two kinds of privacy-protection and switch between them optionally by users: The first is anonymous scheme which can make nobody know the user's identity, including the server and the registration center (RC), and they only know these users are legal or paying members. The other is hiding scheme which owns also privacy-protection property, because the user's identity is not transferred during the process of the proposed protocol, and only the server and the *RC* know the user's identity. About practical environment, we adopt multi-server architecture which can allow the user to register at the *RC* once and can access all the permitted services provided by the eligible servers. Then a new PPS authenticated key agreement protocol is given based on chaotic maps. Security of the scheme is based on chaotic maps hard problems and a secure one way hash function. Compared with the related literatures recently, our proposed scheme can not only own high efficiency and unique functionality, but is also robust to various attacks and achieves perfect forward secrecy. Finally, we give the security proof and the efficiency analysis of our proposed scheme.

Keywords: Chaotic maps, key agreement, multi-server architecture, privacy-protection system

1 Introduction

Authenticated key exchange (AKE) is one of the most important cryptographic components which is used for establishing an authenticated and confidential communication channel. Based on the number of participants, we can divide AKE protocols into three categories: two-party AKE protocols [10], three-party AKE protocols [13], and N-party AKE protocols [3, 14, 25]. Furthermore, based on the respective features in detail, the previous AKE protocols [1, 2, 8, 11, 15, 17, 18, 20, 21, 22, 23] can be classified many categories, we use two-party AKE protocols to set an example: such as password-based [10], chaotic map-based [2], ID-based [25], anonymity [13, 23], secret sharing [21] and so on. Recently many researchers achieve AKE in the multi-server environment called multi-server authenticated key agreement (MSAKA) protocols. MSAKA protocols allow the user to register at the registration center (RC) once and can access all the permitted services provided by the eligible servers. In other words, users do not need to register at numerous servers repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [11] which lead to the fact that user has to register to different servers separately. About MSAKA protocols, the pioneer work in the field was proposed by Li et al. [15] in 2001. However, Lin et al. [17] pointed out that Li et al.s scheme takes long time to train neural networks and an improved scheme based on ElGamal digital signature and geometric properties on the Euclidean plane has also been given. Next stage, the main work is amended repeatedly. For example, Tsai [22] proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsais scheme only uses the nonce and one-way hash function, the problems associated with the cost of computation can be avoided in the distributed network environment. However, the literature [20] pointed out that Tsais scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy. At the present stage, the research emphasis shifts to functionality and user experi-

ence. Therefore, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently [20].

However, there are many scenes need not mutual authentication at all and we just need one-way authentication. For example, readers act upon the perceived reputation of a news source, so reputation is a valuable commodity for journalists. No further authentication is required and since the information is public, channel secrecy is not required and does not affect the actions of either party. Another example, on Internet, patients requiring medical advice may wish to do so anonymously, while still ensuring the confidentiality of their request and assurance that the medical advice received comes from an authentic, qualified source. The key idea of one-way AKE is that one party wishes for no one to be able to determine his/her identity, including all the authorities. However, only a few protocols have considered the problem of one-way authentication. Goldberg [8] gave a specialized one-way AKE security definition for the Tor authentication protocol. The literature [1] described an identity-based anonymous authenticated key exchange protocol but with a limited session key secrecy definition based on key recovery, not indistinguishability. Morrissey et al. [18] analyzed the security of the Transport Layer Security (TLS) protocol in the context of one-way authentication, but with specialized security definitions. Recently, Goldberg and Stebila [9] provided an intuitive set of goals and present a formal model that captures these goals. Usually, public key encryption can be used for one-way AKE protocols, for example by having the client encrypt a session key under the server's public key. This mechanism is widely used, for example in the RSA-based cipher suites in TLS [6] and in the KAS1 protocol in NIST SP800-56B [19].

All above-mentioned scenes do not include a new scene of application: A user wants to consult with an authenticated expert anonymously or explicitly, and the expert does not want to provide the free service because of limited time or energy. Both mutual authenticated key agreement [10] and one-way authenticated key agreement protocol [27] cannot provide the solutions about this scene. Even for mutual authenticated key agreement protocol with privacy protection cannot solve it, because the scene needs transformation flexibly between anonymity and hiding identity. Therefore I propose the concept about privacy-protection system to solve the problem. In a meaning, the mutual authenticated key agreement protocol with privacy protection is the subset of the privacy-protection system.

The main contributions are shown as below: The paper firstly presents a new provable privacy-protection system towards multi-server architecture. Furthermore, the proposed protocol is mainly based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve. In Security aspect, the protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc. About functionality, the protocol also has achieved some well-known properties, such as perfect forward secrecy and execution efficiency. The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a privacy-protection system towards multi-server architecture is described in Section 3. Then, the security analysis and efficiency analysis are given in Section 4 and Section 5. This paper is finally concluded in Section 6.

1.1 Multi-server Architecture

In the multi-server environment [15], each user must perform authentication procedure to login the server for a transaction. If the user is in a single authentication architecture, then the user must register at various servers and memorize the corresponding identifications and passwords, which could not be convenient for a user. In order to make the registration to various servers easier for users, each user must register with the registration center to obtain a secure account. Then the user uses the secure account to perform the login and authentication procedures with various servers.

1.2 Security Requirements

Secure communication schemes for remote one-way authentication and session key agreement for the multi-server architecture should provide security requirements [20, 27]:

1) Authentication: Anonymous authentication or hiding identity authentication in different phase in our protocol. Anonymous authentication: the server or a expert knows that he serves for a premium user but does not know the user's identity. Hiding identity authentication: only the *RC* and the server know the user's identity.

2) Impersonation attack: An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

3) Man-in-the-middle attack: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

4) Replay attack: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

5) Known-key security: Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

Table 1: Notations

Symbol	Definition
SID_A	A temporary session;
S_i, ID_{S_i}	The i^{th} server and the identity of the i^{th} server, respectively;
Ano_{S_i}	The identifier of anonymity;
a, r_a, r_i	Nonces;
$(x, T_k(x))$	Public key based on Chebyshev chaotic maps;
k	Secret key based on Chebyshev chaotic maps;
RC, ID_{RC}	Registration center and its identity;
H	A secure one-way hash function;
$\|\|$	Concatenation operation.

6) Perfect forward secrecy: An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node secret keys cannot results in the compromise of previously established session keys.

7) Session key security: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

8) Resistance to stolen-verifier attacks: An adversary gets the verifier table from servers or RC by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

9) No verification table: there is no verification table at the RC or the server at all.

10) Securely chosen password and time synchronization: Guarantee securely chosen password and no need for time synchronization among parties.

1.3 Kinds of Authentication

Anonymity ensures that a user may use a resource or service without disclosing the user identity completely.

ID hiding usually means that a user may use a resource or service without disclosing the user identity during the protocol interaction, which is a kind of privacy protection partly. A pseudonym is an identifier of a subject other than one of the subject real names. ID hiding usually uses pseudonym to realize. Because the server may store the user identity.

OTP (one-time password) usually means that the password can be used only once but the ID is plaintext during the protocol interaction, so there is no privacy protection.

The above-mentioned terms related with authentication called anonymous authentication, hiding identity authentication and OTP authentication.

2 The Proposed Privacy-Protection System with Multi-Server Architecture

In this section, under the multi-server architecture, a chaotic maps-based one-way authentication key agreement scheme is proposed which consists of five phases: server registration phase, user registration phase, Anonymous authenticated key agreement phase, Hiding identity authenticated key agreement phase, Password changing phase.

2.1 Notations and Chebyshev Chaotic Maps

In this section, any server i has its identity ID_{S_i}. Only RC has its identity ID_{RC} and public key $(x, T_k(x))$ and a secret key k based on Chebyshev chaotic maps and a secure one-way hash function $H(\cdot)$. The concrete notations used hereafter are shown in Table 1.

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 $T_n(x) = \cos(ncos^{-1}(x))$ [24]. Chebyshev polynomial map T_n: $R \rightarrow R$ of degree n is defined using the following recurrent relation:

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

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

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

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

$$T_r(T_s(x)) = T_{rs}(x).$$

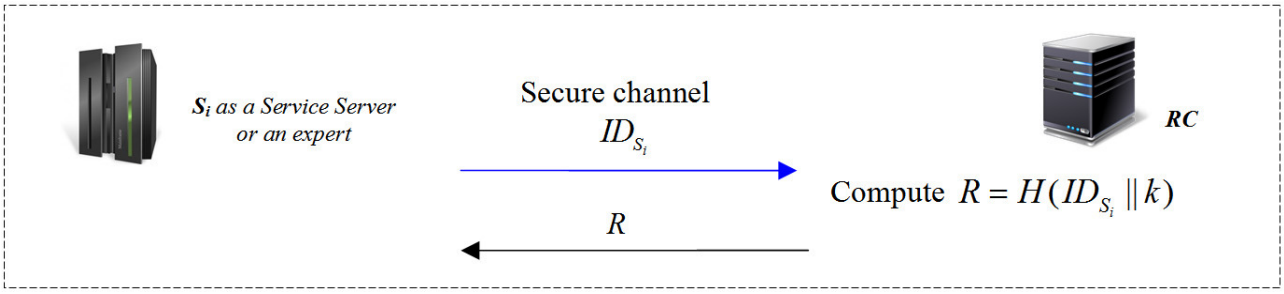


Figure 1: Server or a authenticated expert registration phase

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

$$T_r(T_s(x)) = T_s(T_r(x)).$$

In order to enhance the security, Zhang [26] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

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

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

$$T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x)).$$

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

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

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).*

2.2 Server Registration Phase

The business architecture of our proposed protocol: (1) The *RC* is a platform for users and servers/experts. In other words, anyone can register at the *RC* as a user or an expert. (2) If a user wants to consult with an expert, the *RC* must help him find an authenticated expert and charge fee. After ending the consultation, the user will give the evaluation for the expert, and the *RC* and the expert will share the fee in some percentage. (3) The expert must be authenticated by real name. (4) The user can consult with an expert anonymously or not. (5) Accumulative assessment will affect the expert's reputation.

Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in multi-server architecture, it is assumed that the servers can register at the registration center in some secure way or by secure channel. The same assumption can be set up for servers. Figure 1 illustrates the server registration phase.

Step 1. When a server(or an expert) wants to be a new legal service provider, she chooses her identity ID_{S_i} with her identification card in law. Then the server submits ID_{S_i} to the *RC* via a secure channel.

Step 2. Upon receiving ID_{S_i} from the server, the *RC* computes $R = H(ID_{S_i}||k)$, where k is the secret key of *RC*. Then the server stores R in a secure way via a secure channel.

2.3 User Registration Phase

Figure 2 illustrates the user registration phase.

Step 1. When a user wants to be a new legal user, she chooses her identity ID_A, a random number r_a, and computes $H(r_a||PW)$. Then Alice submits $ID_A, H(r_a||PW)$ to the *RC* via a secure channel.

Step 2. Upon receiving $ID_A, r_a, H(r_a||PW)$ from Alice, the *RC* computes $B = H(ID_A||k) \oplus H(r_a||PW)$ and $B_A = H(Anonymous||k) \oplus H(r_a||PW)$, where k is the secret key of *RC*. Then Alice stores $\{ID_A, r_a, B, B_A\}$ in a secure way.

2.4 Anonymous Authenticated Key Agreement Phase

In this phase, the anonymous authentication has three meanings: (1) The server and the *RC* authenticated each other; (2) The *RC* will help the server to authenticate the premium user, but no one knows (including the server and the *RC*) the premium user's identity. (3) The *RC* will help the premium user to authenticate the server. This concrete process is presented in Figure 3.

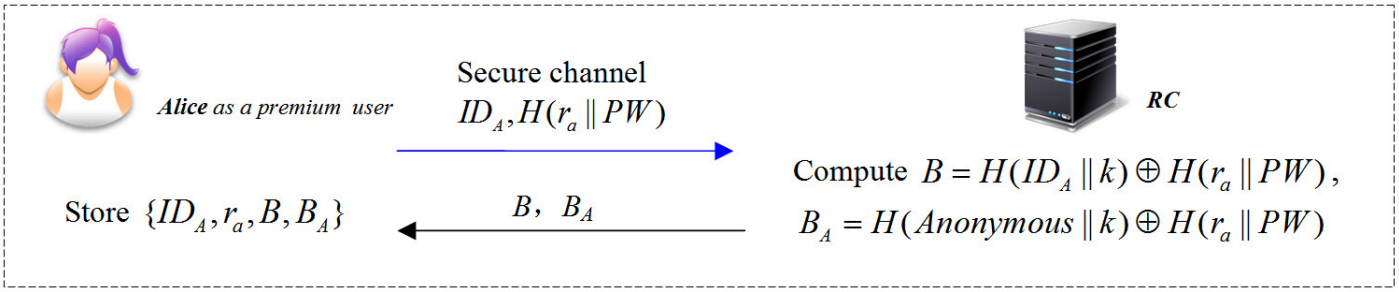


Figure 2: The user registration phase

Step 1. If Alice(assume Alice as a premium user) wishes to consult some personal issues establish with S_i (or an expert) in a anonymous way, she will input password and compute $B_A^* = B_A \oplus H(r_a||PW)$, n choose a random integer number a and compute $K_{A-RC} = T_a T_k(x)$, $H_A = H(B_A^*||ID_{S_i}||T_a(x))$. After that, Alice sends $m_1 = \{Ano_{S_i}, T_a(x), H_A\}$ to S_i where she wants to get the server's service.

Step 2. After receiving the message $m_1 = \{Ano_{S_i}, T_a(x), H_A\}$ from Alice, S_i will do the following tasks to ask RC for helping Alice to authenticate itself: S_i selects random r_i and computes $T_{r_i}(x)$ and $C_1 = H(ID_{S_i}||m_1||R||T_{r_i}(x))$. And then sends the message m_2 to RC.

Step 3. Next, RC will help Alice to authenticate S_i and verify the temporary information by helping them to compute the session key. After receiving the message $m_2 = \{ID_{S_i}, T_{r_i}(x), C_2, m_1\}$, RC will do the following tasks:

1) Authenticate S_i; Based on ID_{S_i}, RC can compute $R^{'} = H(ID_{S_i}||k)$. Then RC computes $C_1^{'} = H(ID_{S_i}||m_1||R^{'}||T_{r_i}(x))$ and check if $C_1^{'}? = C_1$. If above equation holds, that means S_i is legal participant in this instance because only S_i owns R.
2) Anonymous authenticate Alice: RC computes $B_A^* = H(Anonymous||k)$, $H_A^{'} = H(B_A^*||ID_{S_i}||T_a(x))$ and verifies if $H_A^{'} = H_A$ holds. If above equation holds, that means Alice is a legal premium user in this instance because only a legal premium user can retrieve the information $H(Anonymous||k)$.
3) Confirm S_i is the server that Alice wants to consult with: RC computes computes $H_A^{'} = H(B_A^*||ID_{S_i}||T_a(x))$. RC verifies $H_A^{'}? = H_A$. If holds, that means S_i is the server that Alice wants to consult with.
4) Help S_i and Alice to get the session key: RC computes $C_2 = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and $C_3 = H(B_A^*||ID_{S_i}||ID_{RC}||T_{r_i}(x))$. Then RC sends the message $\{ID_{RC}, C_3\}$ to Alice and sends the message $\{ID_{RC}, C_2\}$ to S_i.

If any authenticated process does not pass, the protocol will be terminated immediately.

Step 4. For Alice: After receiving the message $\{ID_{RC}, C_3\}$, Alice computes $C_3^{'} = H(B_A^*||ID_{S_i}||ID_{RC}||T_{r_i}(x))$. Check if $C_3^{'} = C_3$. If holds, Alice computes $SK = T_a T_{r_i}(x)$.

For S_i: After receiving the message $\{ID_{RC}, C_2\}$, S_i computes $C_2^{'} = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and checks if $C_2^{'} = C_2$. If holds, then S_i computes $SK = T_{r_i} T_a(x)$.

Remark 1: We can view the servers and the RC as an integrated system for the user, so from the perspective of the user, we adopt anonymous authentication, that means only user authenticated the integrated system (the server and the RC) but there is a anonymous authentication for the user. However, from the inside integrated system, for providing the reliable service in multi-server architecture, and we must make the server and the RC to authenticate each other, that is the mutual authentication.

2.5 Hiding Identity Authenticated Key Agreement Phase

Simply speaking, a premium user also can as a legal and hiding ID to interact with an expert. The two differences between hiding identity authenticated and anonymous authenticated are:

1) The user uses the B to login at the RC so that the server or the expert can know the users positive identity.
2) We construct an efficient method to covered identity or some important information instead of using symmetric cryptography. Without loss of generality, we assume Party i sends a covered message to Party j using $(x, T_{K_j}(x))$ for covering ID_i but only Party j can recover the ID_i. Party i selects a large and random integer t, and computes $T_t(x)$, $C_t = T_t T_{K_j}(x) ID_i$, $H(C_t||T_t(x))$.

   Then Party i sends $\{T_t(x), C_t, H(C_t||T_t(x))\}$ to Party j. After receiving the message $\{T_t(x), C_t, H(C_t||T_t(x))\}$ from Party i, Party j will use $T_t(x)$

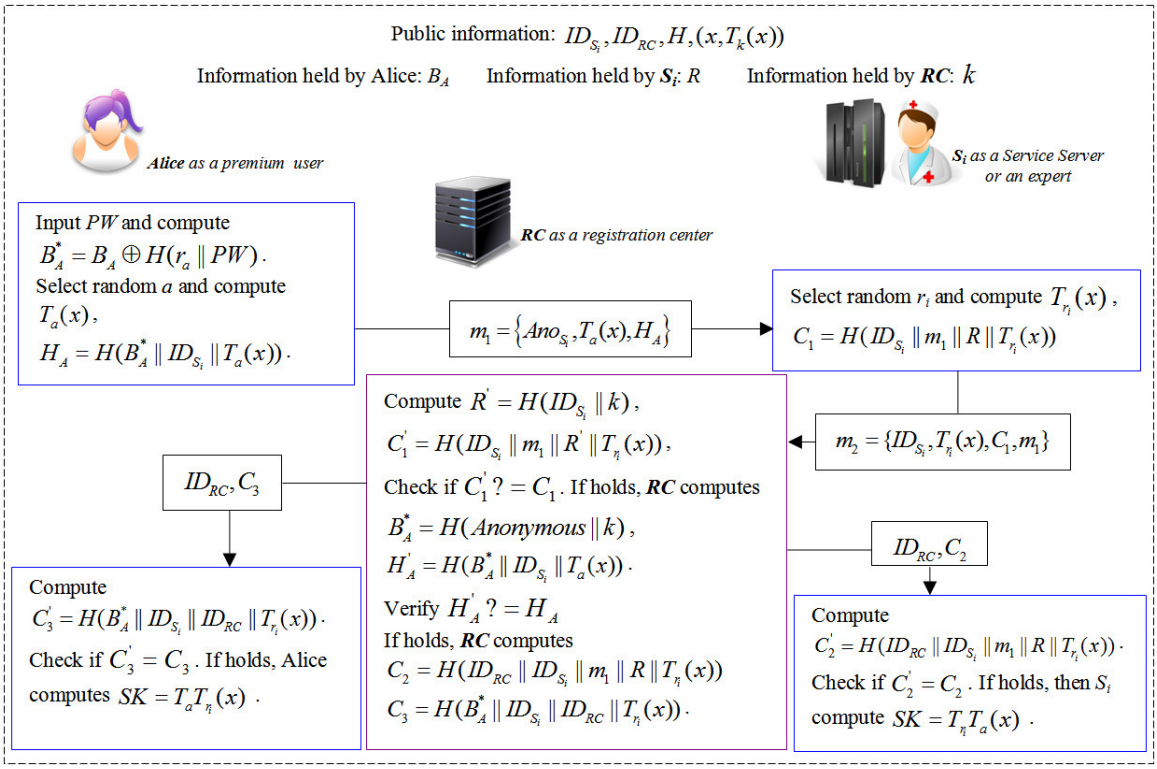


Figure 3: Anonymous authenticated key agreement phase for multi-server environment

and his own secret key K_j to recover $ID_i = C_t/T_{K_j}T_t(x) = C_t/T_tT_{K_j}(x)$. Then Party j check if the two hash values are equal. If above equation holds, Party j deems ID_i is legal identity. Otherwise, Party j terminates the session.

For the sake of simplicity, the paper only provides the process of hiding identity authenticated key agreement phase (Figure 4).

2.6 Password Changing Phase

Figure 5 illustrates the password changing phase.

Step 1. When a user wants to change her password, she chooses new password $PW^{'}$, two random numbers $r_a^{'}$, a, and computes $B^* = B \oplus H(r_a||PW)$, $T_a(x)$, $K_{A-RC} = T_aT_k(x)$, $H_A = H(B^*||ID_{RC}||T_a(x)||C_1||C_2)$, $C_1 = ID_A \times K_{A-RC}$ and $C_2 = H(r_a^{'}||PW^{'}) \times K_{A-RC}$. Then Alice sends $m_1 = \{T_a(x), C_1, C_2, H_A\}$ to the RC.

Step 2. Upon receiving $m_1 = \{T_a(x), C_1, C_2, H_A\}$ from Alice, RC computes $K_{RC-A} = T_kT_a(x)$ and recovers $ID_A = C_1/K_{RC-A}$, $H(r_a^{'}||PW^{'}) = C_2/K_{RC-A}$. Next, RC computes $B^* = H(ID_A||k)$ and $H_A^{'} = H(B^*||ID_{RC}||T_a(x)||C_1||C_2)$. Then, RC checks $H_A^{'} = H_A$ or not. If holds, RC computes

$$\begin{aligned} B^{'} &= H(ID_A||k) \oplus H(r_a^{'}||PW^{'}), \\ B_A^{'} &= H(Anonymous||k) \oplus H(r_a^{'}||PW^{'}), \\ H_{RC} &= (ID_{RC}||ID_A||B^{'}||B_A^{'}), \\ C_3 &= B^{'} \times K_{RC-A}, \\ C_4 &= B_A^{'} \times K_{RC-A}, \end{aligned}$$

where k is the secret key of RC. Finally RC sends $\{ID_{RC}, C_3, C_4, H_{RC}\}$ to Alice.

Step 3. Upon receiving $\{ID_{RC}, C_3, C_4, H_{RC}\}$, Alice uses K_{A-RC} to decrypt C_3, C_4 to get $B^{'}, B_A^{'}$. Then Alice computes locally $H_{RC}^{'} = (ID_{RC}||ID_A||B^{'}||B_A^{'})$ to compare with H_{RC}. If they are equal, Alice stores $\{ID_A, r_a^{'}, B^{'}, B_A^{'}\}$ in a secure way.

3 Security Analysis

The section analyzes the security of our proposed protocol. Let us assume that there are three secure components, including the two problems CMBDLP and CMBDHP cannot be solved in polynomial-time and a secure one-way hash function. Assume that the adversary has full control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages.

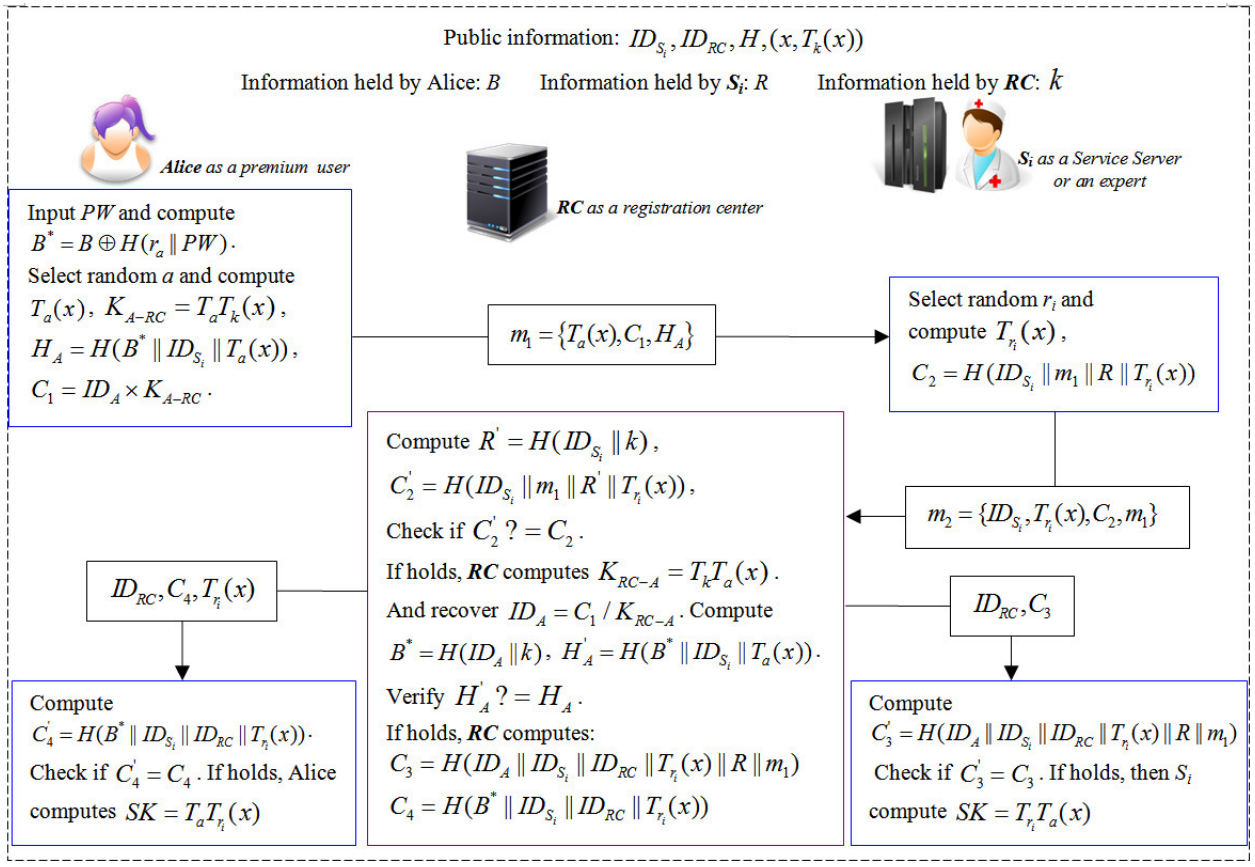


Figure 4: Hiding identity authenticated key agreement phase for multi-server environment

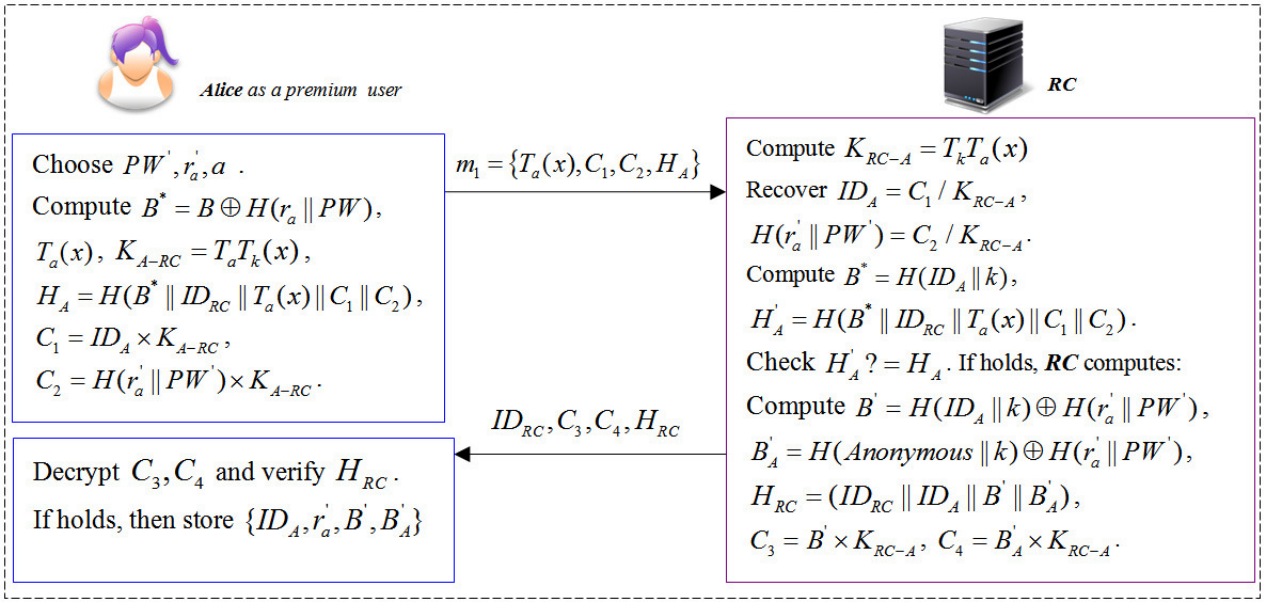


Figure 5: The password changing phase

3.1 Security Proof of the Proposed Scheme

In this subsection, we give a definition and simplified proof of various kinds of security and attacks.

Anonymous Authentication and Key Agreement.

Definition: Anonymous authentication and key agreement refers to authenticate each other for two peers/system, but only one peer knows the other peer's identity and getting the session key simultaneously.

Simplified Proof: Alice authenticates RC: Because only RC has the secret k, RC can computes $K_{RC-A} = T_k T_a(x)$ which equals to $K_{A-RC} = T_a T_k(x)$. So if Alice computes C_3 and check if $C_3' = C_3$. RC and S_i authenticate each other: We can use the shared key R to achieve the task. Firstly, based on ID_{S_i}, RC can compute $R' = H(ID_{S_i}||k)$ by its private key k. Then RC computes $C_1' = H(ID_{S_i}||m_1||R'||T_{r_i}(x))$ and checks if $C_1' = C_1$. If above equation is equal, then that means RC authenticates S_i. After receiving the messages $\{ID_{RC}, C_2\}$, S_i computes $C_2' = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and checks if $C_2' = C_2$. As for the key agreement, after authenticating each other, the temporary $T_a(x)$, $T_{r_i}(x)$ and the $SID_A||ID_{S_i}||ID_{RC}$ were already authenticated by RC. So finally Alice and S_i can make the key agreement simultaneously. The hiding identity authenticated key agreement can be proof in some analogous way.

Impersonation Attack.

Definition: An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

Simplified Proof: An adversary cannot impersonate anyone of the S_i and RC. The proposed scheme has already authenticated each other between S_i and RC, and Alice authenticates S_i and RC based on the secrets k, R and the nonces a, r_i. So there is no way for an adversary to have a chance to carry out impersonation attack.

Man-in-the-middle Attack.

Definition: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

Simplified Proof: Because $C_i (1 \leq i \leq 3)$ contain the participants' identities or an anonymous identifier, a man-in-the-middle attack cannot succeed.

Replay Attack.

Definition: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

Simplified Proof: That any message of Alice was replayed by an adversary is meaningless. Because "Alice" is an anonymous user, the adversary can as an anonymous user to initiate the protocol legally as his wish. Furthermore, if the adversary wants to launch the replay attack successfully, it must compute and modify $T_a(x)$, $T_{r_i}(x)$ and $C_i (1 \leq i \leq 3)$ correctly which is impossible.

Known-key Security.

Definition: Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

Simplified Proof: Since the session key is depended on the random nonces a and r_i, and the generation of nonces is independent in all sessions, an adversary cannot compute the previous and the future session keys when the adversary knows one session key.

Perfect Forward Secrecy.

Definition: An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node's secret keys cannot results in the compromise of previously established session keys.

Simplified Proof: In the proposed scheme, the session key is related with a and r_i, which were randomly chosen by Alice and the server S_i respectively. So any session key has not related with the secret key (such as k) of each of participants. Furthermore, because of the intractability of the CMBDLP and CMBDHP problem, an adversary cannot compute the previously established session keys.

Session Key Security.

Definition: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

Simplified Proof: In the authenticated key agreement phase, a session key SK is generated from a and r_i. These parameter values are different in each session, and each of them is only known by Alice and S_i. Additionally, since the values

Table 2: Security of our proposed protocol

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
[5] (2013)	Yes	S21	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
[22] (2008)	Yes	S21	Yes	Yes	Yes	No	No	No	No	No	Yes	No
[16] (2009)	Yes	No	Yes	Yes	Yes	S61	No	No	No	Yes	Yes	No
[7] (2009)	Yes	No	Yes	No	Yes	S61	No	No	No	Yes	Yes	No
Our Scheme	Yes	S22	Yes	Yes	Yes	S62	Yes	Yes	Yes	Yes	Yes	Yes

S1: Single registration; S2: Authentication; S21: Mutual Authentication; S22: Privacy-protection system; S3: No verification table; S4: Securely chosen password; S5: Session key agreement; S6: Privacy protection for a user; S61: ID hiding; S62: Anonymity or ID hiding; S7: Freedom from time synchronization; S8: Session key secrecy; S9: Perfect forward secrecy; S10: Resistance to replay attack; S11: Resistance to stolen-verifier attack; S12: Resistance to masquerading attack Yes/No: Support/Not support the security.

Table 3: Descriptions the model of Canetti and Krawczyk

Symbol	Definition
Parties $P_1, \cdots, P_n$	Modelled by probabilistic Turing machines.
Adversary Λ	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

a and r_i of the random elements are very large, attackers cannot directly guess the values a and r_i of the random elements to generate session key. Therefore, the proposed scheme provides session key security.

Resistance to Stolen-verifier Attacks.

Definition: An adversary gets the verifier table from servers or RC by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

Simplified Proof: In the proposed scheme, neither the server nor the registration center maintains any verification table. Thus, the stolen-verifier attack is impossible to initiate in the proposed scheme.

From Table 2, we can see that the proposed scheme can provide secure session key agreement, perfect forward secrecy and so on. As a result, the proposed scheme is more secure and has much functionality compared with the recent related scheme.

Remark 2: Some qualitatively discuss about the difference between the proposed scheme and [5, 7, 16, 22] as followed: (1) Our protocol is one way authentication AKE for users, so only servers need to registration at the RC. (2) About authentication, one-way authentication for users and mutual authentication for server and RC (see Remark 1). (3) Our proposed protocol can hold the security S1-S12, but the [5, 7, 16, 22] have some defects. (4) Our protocol is anonymity, and [7, 16] only assure ID hiding, and [5, 22] have no privacy protect at all.

3.2 The Provable Security of the Proposed Scheme

We recall the definition of session-key security in the authenticated-links adversarial model of Canetti and Krawczyk [4]. 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(P_i):This query allows the adversary to take over 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 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(P_i) before s is expired.

Definition 4. *An authenticator exchange protocol Π_1 in security parameter k is said to be authentication secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,*

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

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

Theorem 1. *Under the CMBDHP assumption, using Algorithm 1 to compute two authenticator messages can be deemed as session keys which are session-key secure in the adversarial model of Canetti and Krawczyk [4].*

Proof. The proof is based on the proof given by [4]. There are two-two uncorrupted parties (Alice and the server, Bob and the server) in matching sessions output the same authenticator messages, and thus the first part of Definition 4. 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. □

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.

Definition 5. *A composable key exchange protocol Π_2 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,*

1) If two uncorrupted parties have completed matching sessions with pre-distributed parameter, these sessions produce the same key as output;

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

Theorem 2. *Under the CMBDHP assumption, using Algorithm 2 to compute session key is session-key secure in the adversarial model of Canetti and Krawczyk [4].*

Proof. The proof's process is similar to Theorem 1. The protocol Π_2 is the composable instance of protocol multiple Π_1. Since Theorem 1 is session-key secure, the protocol Π_2 is also session-key secure. □

Probability analysis. It is similar to Algorithm 1. If we assume that Algorithm 2 forms a polynomial-time distinguisher for CMBDHP having non-negligible advantage, the overall advantage of the proposed protocol simulator with authenticated parameter is ε/k which is also non-negligible. Because the protocol Π_2 chooses different parameters to structure session keys in different phase which are secure independence of protocol Π_1.

4 Efficiency Analysis

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, Wang [24] proposed several methods to solve the Chebyshev polynomial computation problem. For convenience, some notations are defined as follows.

T_{hash}: The time for executing the hash function;

T_{sym}: The time for executing the symmetric key cryptography;

T_{XOR}: The time for executing the XOR operation;

T_{Exp}: The time for a modular exponentiation computation;

T_{CH}: The time for executing the $T_n(x) \, modp$ in Chebyshev polynomial using the algorithm in the literature [12].

To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [12]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

Table 4 shows performance comparisons between our proposed scheme and the literature of [5, 7, 16, 22] in

Algorithm 1 CMBDHP distinguisher

Input: $H, E_K()/D_K(), (x, T_k(x))$

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 a user send $\{i, Ano_{S_i}, T_a(x), H_A\}$ to a server $\boldsymbol{S_i}$, and let the server $\boldsymbol{S_i}$ send $\{i, ID_{S_i}, T_{r_i}(x), C_1, m_1\}$ to the ***RC***, where i is the session identifier. The ***RC*** can compute the encrypted messages $\{C_2, C_3\}$ with the authenticators locally after authenticating the server $\boldsymbol{S_i}$ 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** $d \xleftarrow{R} \{0,1\}$.
8:**end if**
Output: d

Algorithm 2 Proposed protocol simulator

Input: $H, (x, T_a(x)), (x, T_{r_i}(x), (x, T_k(x))$

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, After running the protocol Π_1, the ***RC*** can compute the encrypted messages $\{C_2, C_3\}$ with the authenticators locally. Then the ***RC*** continues to send messages $\{ID_{RC}, C_3\}$ and $\{ID_{RC}, C_2\}$ to the user and the server $\boldsymbol{S_i}$ respectively. Both the user and the server can compute the session key $SK = T_a T_{r_i}(x)$ locally after authenticating each other by ***RC***'s 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** $d \xleftarrow{R} \{0,1\}$.
8:**end if**
Output: d

Table 4: Efficiency of our proposed scheme

Phase	[5] (2013)	[22] (2008)	[16] (2009)	[7] (2009)	Ours
A	$2T_{hash} + 1T_{XOR}$	$2T_{hash} + 1T_{XOR}$	$5T_{hash} + 2T_{XOR}$	$8T_{hash} + 4T_{XOR}$	$3T_{hash}$
B	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$
C	$2T_{hash} + 1T_{XOR} + 1T_{Exp}$	$1T_{hash} + 2T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$7T_{hash} + 7T_{XOR}$	N/A
D1-User	$1T_{hash} + 1T_{Exp}$	$4T_{hash} + 3T_{XOR}$	$3T_{hash}$	$2T_{hash}$	$3T_{hash} + 1T_{CH}$
D1-Server	$2T_{hash} + 2T_{Exp}$	$6T_{hash} + 7T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$8T_{hash} + 6T_{XOR}$	$2T_{hash} + 1T_{CH}$
D1-RC	$6T_{hash}$	$6T_{hash} + 5T_{XOR}$	0	$5T_{hash} + 7T_{XOR}$	$6T_{hash} + 2T_{CH}$
D1-Total	$9T_{hash} + 3T_{Exp}$	$16T_{hash} + 15T_{XOR}$	$9T_{hash} + 3T_{XOR}$	$15T_{hash} + 13T_{XOR}$	$11T_{hash} + 4T_{CH}$
D2-User	N/A	N/A	N/A	N/A	$3T_{hash} + 1T_{CH}$
D2-Server	N/A	N/A	N/A	N/A	$2T_{hash} + 1T_{CH}$
D2-RC	N/A	N/A	N/A	N/A	$6T_{hash} + 2T_{CH}$
D2-Total	N/A	N/A	N/A	N/A	$11T_{hash} + 4T_{CH}$
E	$2T_{hash} + 2T_{XOR}$	$2T_{hash} + 2T_{XOR}$	$4T_{hash} + 5T_{XOR}$	$4T_{hash} + 4T_{XOR}$	$8T_{hash} + 2T_{CH}$
F	4 rounds	7 rounds	3 rounds	5 rounds	3 rounds

A: User registration; B: Server registration; C: Login phase; D1: Hiding identity authentication phase; D2: Anonymous authentication phase; E: Password change phase; F: Communication cost; N/A: No support.

multi-server architecture. Therefore, as in Table 4 the concrete comparison data as follows: The total computation cost of our proposed protocol is lower than the literatures [5]. The main reason is that the literatures [5] adopted modular exponentiation computation. At the same time, the literatures [5] cannot provide privacy protection for a user. The total computation cost of our proposed protocol is higher than the literatures [7, 16, 22]. Furthermore, the communication round of our proposed protocol is superior to the literature [7, 22] and is equal to the literature [16]. The reasons are: one reason is our protocol mainly adopts Chebyshev chaotic maps but the literatures [7, 16, 22] mainly adopts one way hash function. At the same time, Chebyshev chaotic maps has more attributes which leading to reduce communication rounds. Furthermore, from the perspective of security, our protocol is more secure than the literatures [7, 16, 22]. From Table 2, we can see that the literatures [5, 7, 16, 22] cannot resist many attacks and the literatures [7, 16] cannot afford any authentication method. Therefore, as in Table 2 and Table 4, we can draw a conclusion that the proposed scheme has achieved the balance of efficiency and security.

5 Conclusion

We only use chaotic maps and a secure one-way hash function to construct a provable privacy-protection system (PPS) which provides a provable privacy-protection system towards multi-server architecture. The core ideas of the proposed system are the mutual authentication between the servers and *RC* and the anonymity or hiding identity for the users. Subsequently, we explain the practical motivations for authentication and secrecy assurances of parties engaging in AKE protocols and some related terms. Based on our discussion we proposed a suitable protocol that covers those goals and offered an efficient protocol that formally meets the proposed security definition. Finally, after comparing with related literatures (multi-server schemes and privacy-protection system) respectively, we found our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

References

[1] K. Aniket, G. M. Zaverucha, and G. Ian, "Pairing-based onion routing with improved forward secrecy," *ACM Transactions on Information and System Security*, vol. 13, no. 4, pp. 29, 2010.

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

[3] 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.

[4] 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.

[5] T. Y. Chen, C. C. Lee, M. S. Hwang and J. K. Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, 2013.

[6] T. Dierks and A. Christopher, *The TLS Protocol Version 1.0*, RFC 2246, 1999. (`http://www.ietf.org/rfc/rfc2246.txt`)

[7] H. C. Hsiang, W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standard & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[8] G. Ian, "On the security of the Tor authentication protocol," in *Privacy Enhancing Technologies*, LNCS 4258, no. 2, pp. 316–331, Springer, 2006.

[9] G. Ian, D. Stebila, and U. Berkant, "Anonymity and one-way authentication in key exchange protocols," in *Design Codes and Cryptography*, vol. 67, no.5, pp. 245–269, 2013.

[10] J. Katz, P. MacKenzie, G. Taban, V. Gligor, "Two-server password-only authenticated key exchange," in *Applied Cryptography and Network Security*, LNCS 3531, pp. 1-16, Springer, 2005.

[11] M. K. Khan, J. Zhang, "Improving the security of a flexible biometrics remote user authentication scheme," *Computer Standard & Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

[12] L. Kocarev, S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, pp. 53–54, Springer, 2011.

[13] C. C. Lee, C. T. Li, C. W. 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.

[14] H. Li, C. K. Wu, J. Sun, "A general compiler for password-authenticated group key exchange protocol," *Information Processing Letters*, vol. 110, no. 4, pp. 160–167, 2010.

[15] L. H. Li, I. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[16] Y. P. Liao, S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multiserver environment," in *Computer Standard & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[17] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[18] P. Morrissey, N. P. Smart, and B. Warinschi, "A modular security analysis of the TLS handshake protocol," in *Advances in Cryptology*, LNCS 5350, pp. 55–73, Springer, 2008.

[19] NIST, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, NIST National Institute of Standards and Technologe, pp. 800, 2009.

[20] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[21] M. D. Raimondo, R. Gennaro, "Provably secure threshold password-authenticated key exchange," in *Journal of Computer and System Sciences*, vol. 72, no. 6, pp. 978–1001, 2006.

[22] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no. 3-4, pp. 115–121, 2008.

[23] H. Tseng, R. Jan, W. Yang, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *IEEE International Conference on Communications (ICC'09)*, pp. 1–6, 2009.

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

[25] 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.

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

[27] H. F. Zhu, "A provable one-way authentication key agreement scheme with user anonymity for multi-server environment," *KSII Transactions on Internet And Information Systems*, vol. 9, no. 2, pp. 811–829, 2015.

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.

Yifeng Zhang, 24 years old, an undergraduate from Shenyang Normal University, major in information security management. During the four years of college, after completing her studies, he enjoys reading the book related to this major. Under the guidance of the teacher, he has published six articles in EI journals.

Yang Sun obtained his master degree in Information Science and Engineering from Northeastern University. Yang Sun is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a department head of network engineering. He has research interests in wireless networks, mobile computing, cloud computing, social networks and network security. Yang Sun had published more than 15 international journal and international conference papers on the above research fields.

Composable Secure Roaming Authentication Protocol for Cloud-assisted Body Sensor Networks

Qing-Qing Xie[1], Shun-Rong Jiang[2], Liang-Min Wang[1], and Chin-Chen Chang[3,4]
(Corresponding author: Chin-Chen Chang)

School of Computer Science and Technology, Anhui University[1]
NO. 111, Jiu Long Rd., Hefei, Anhui 230601, China
School of Computer Science and Communication Engineering, Jiangsu University[2]
Zhenjiang, Jiangsu, 212013, China
Department of Information Engineering and Computer Science, Feng Chia University[3]
No. 100, Wenhwa Rd., Seatwen, Taichung, Taiwan 40724 (R.O.C.)
Department of Computer Science and Information Engineering, Asia University[4]
No. 500, Lioufeng Rd., Wufeng, Taichung, Taiwan 41354 (R.O.C.)
(Email: alan3c@gmail.com)
(Received May 23, 2014; revised and accepted Sept. 12 & Nov. 24, 2014)

Abstract

The cloud-assisted Body Sensor Networks (BSN) often has an architecture of Multi-hop Wireless Networks (MWN) model, in which both the body sensors and the users must be secure to protect the whole infrastructure. Unfortunately, both the information providers and the users are movable and resource-constrained in communication and computation. Thus some new security problems are proposed, such as the light weight-secure authentication caused by limited resource, re-authentication in foreign zone caused by mobility, and composablilty security caused by heterogeneity between the transmission subnet, many BSN subnets. We propose a Random Roaming Authentication Protocol (RanRAP) for BSNs with these cloud-assisted infrastructure. We test the composable security at an AP/cluster head/gateway node by using strand spaces theory, and analyze the performance of RanRAP protocol in both the theoretical analysis and experiment simulations. It was shown that RanRAP has some advantages of composable security, computation and communication overheads over some related protocols.

Keywords: Authentication protocol, body sensor networks, composable security, internet of things

1 Introduction

Body Sensor Networks (BSNs) [4, 14] have emerged as a promising technology for medical and non-medical applications, which are also called Wireless Body Area Sensor Networks (WBANs). BSNs consist of a number of miniaturized, portable, and autonomous sensor nodes that are used to monitor the body function and the surrounding environment. These sensor nodes continuously collect vital signs of patients, which are used for ubiquitous health monitoring including real-time diagnosis and prescription. In addition, BSNs may be used for managing catastrophic events and increasing the effectiveness and performance of rescue forces. The huge amount of data collected by WBAN nodes should be saved and preceded in a scalable, on-demand, powerful, and secure manner. Cloud-assisted BSNs are emerged and expected to satisfy these requirements [8]. Typical Cloud-assisted BSN works in the architecture of Multi-hop Wireless Network (MWN) Model [24], [25] as shown in Figure 1, in which a backbone transmission subnet is employed to connect the BSN clusters with cloud.

In Figure 1, the sensor clusters are formed by the body sensors located in a near place. These sensors are weak in computing and communication, but they are movable with the worn person. Thus the sensor can roam from cluster A to cluster B. The transmission subnet is a fixed infrastructure, e.g., the internet, wire networks, and some other steady wireless devices connected to a powerful cloud computing center. Each cluster has an access point (AP) to the backbone network. The AP is powerful in computation and communication, and also serves as the head of the cluster (CH) and the gateway node of the BSN cluster. We take each cluster as a BSN subnet, and the cluster head (CH, also AP) as the base station of the located subnet. Then BSNs with the cloud-assisted

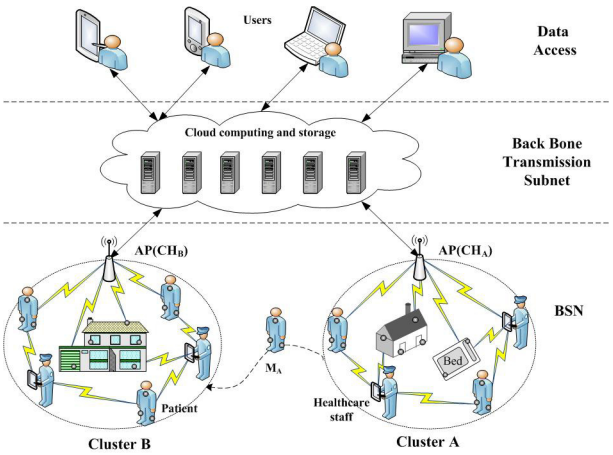


Figure 1: Cloud-assisted BSNs

infrastructure become scalable, for each AP has a cluster and the number of the AP is assumed no limitation in the fixed transmission subnet. The storage and processing of big data collected by BSN isn't a problem, either. Because AP sends the collected information to the cloud computing center through the transmission backbone network, and the cloud will save and process the big data.

Now we discuss the security of cloud-assisted BSN. We won't consider the internal security questions in a BSN cluster, the transmission backbone network and the cloud center with the assumption that they are solved in their areas respectively. We focus on the new security question on the node roaming authentication. For example, node M_A moves away from his home cluster A, and joins a foreign cluster B. M_A cooperates with the node in the cluster B, and the BSN also requires CH_B to collect continually the information of M_A.

In the remainder part of this paper, we will study the roaming identity authentication between CH_B and M_A. This identity authentication has several features. The first is the requirement of the lightweight. The mobile nodes in the BSN are resource-limited in communication, computation and even power supply. Secondly, the selection of the visiting foreign cluster is usually on demand and random. That is to say, a node often moves from one cluster to another in a random manner after the node registers in the initial home cluster. Besides, the node is unwilling to go back to the home cluster to obtain the trusted recommendation certification before joining a new cluster. Finally, when the mobile nodes join in a foreign cluster and obtain their legal identities, they want to access all the network resources of the foreign cluster. Therefore, the authentication protocol should have both the self-security and the composable security [30, 31], and shouldn't affect the security of the existed protocols in the foreign cluster. A typical composable security scenario is that the running of the identity authentication protocol in the cluster head shouldn't reduce the security of other protocols running in that head. Currently, papers about the roaming authentication protocol for this Cloud-assisted BSN architecture are very limited. Up to now, there are no references on the authentication protocol with the random roaming and composable security.

2 Related Work

It seems that the traditional research area of the Secure Wireless Roaming [12, 33, 34, 39] is related to our topic. However, these protocols are realized by the session keys negotiation based on the public key mechanisms. The computation of the public key algorithm costs too much. Thus it is difficult to complete the computation in the node of the BSN. There is also no composable and secure protocol in this area.

The universal composable (UC) security [30, 31] refers to the situation that the protocol guarantees the security when it is in the following cases: running alone, composed of an arbitrary set of protocols, and more generally, used as a component of an arbitrary protocol. Some protocols [1, 3, 19] are designed or analyzed by using the UC formal approach. Unfortunately, the present formal protocol design method for the composable security is combined with a strong security, which fails to guarantee the lightweight property. Later UC security is integrated into the design of roaming authentication protocol, such as [7], which, however, did not attend to the lightweight property.

The typical lightweight authentication protocols in the area of wireless sensor networks is proposed by Perrig et al. [28] who presented the lightweight secure structure SPINS and the broadcast authentication protocol μTESLA. The μTESLA used a reverse hash chain to replace the public-cryptography-based heavy algorithms. Du et al. [6] reduced the computation and communication overheads by adopting the Merkle Tree to construct an authentication path. Further, the whole network was divided into some subareas to reduce the Merkle Tree height, and protocol authentication hops were also reduced. Only the static nodes were considered in [6, 28]. Later many security studies took mobile nodes into consideration, such as the mobile authentication [2, 18, 38], and roaming authentication for wireless communication [11, 23, 36]. But they are not lightweight enough for wireless sensor networks.

The most related work of BSN security is reported in [13, 14, 21, 29]. Huang et al. [14] and Li et al. [21] present a survey on secure access and data security respectively, but they didn't talk about the roaming authentication. References [13, 29] are discussing the lightweight roaming authentication schemes for the wireless sensor networks. Han [13] considered the re-authentication issue on the mobile nodes moving among different sink nodes. The sink in the home cluster is assumed as a trusted third party, and the adjacent relation of the clusters is assumed as the pre-known information. Then the authentication materials are pre-stored in the adjacent clusters. Thus the credible information is also assumed to be transferred

to the adjacent clusters. That is to say, the foreign cluster is limited as one of a neighbor of the home cluster. In this way, the communication and computation expenses of the re-authentication are reduced by the neighbor roaming assumption and pre-transferred information. Here this binding relation of neighboring clusters loosed for the cluster heads are connected by the fixed infrastructure of the transmission subnet.

Qiu [29] presented a roaming authentication protocol, in which a mobile node roams within a very large and distributed wireless sensor network, such as the application of the BSN in the healthcare field. When the dynamic sensor moves to a new area (foreign cluster), it sends a request message to the base station before connecting with the router (cluster head) of the area. After verifying the validity of the request message, the base station generates the session key for the mobile node and the router, and sends it to the router. Then the router sends the material of the session key to the mobile node. In Qiu's protocol, the overheads of the base station are too heavy and the communication band near the base station becomes the bottleneck of the system.

We also studied the re-authentication protocol in heterogeneous wireless sensor networks with some mobile sink. In literature [16], we focus on the wireless sensor network based on the classic structure of Voronoi graph, and deduce the computation and storage cost of the presented protocol by using the knowledge of Voronoi graph. In literature [17], we consider a mobile wireless network with a base station, which presented as an on-line trusted authority.

But the scenario of this paper is different from the reported work. At first, there is no base station on-line taking care of the body sensors in networks. Furthermore, the BSN based on a cloud-assisted infrastructure has a MWN model, in which the communication among routers(Cluster Heads) is transferred to the transmission subnet and the computation of the base station was run by the cloud computing center. Thus, the main contribution of this paper is that we focus on a new case that the BSN is connected with a cloud computing center and a backbone transmission network. The presented RanRAP satisfies the random roaming, lightweight and composable security. To the best of our knowledge, our RanRAP is the first reported authentication protocol for the roaming scenario of the presented cloud-assisted BSN.

3 Roaming Authentication Protocol

Our *RanRAP* can be divided into two phases, Phase 1 and Phase 2. In Phase 1, the mobile node registers in the initial home cluster. The secret materials are set and preloaded on the mobile node, such as initial key and authentication information. In Phase 2, the mobile node and the foreign cluster head authenticate each other, and then generate a session key.

Table 1: Notation and description

Notation	Description
t_i	Timestamp
M_A	Mobile node
CH_A	Home cluster
CH_B	Foreign cluster
K_{A_B}	Session key between A and B
$E_K(\cdot)$	Encrypt the plaint message by K
$D_K(\cdot)$	Decrypt the ciphertext by K
$MAC_K(\cdot)$	Message authentication code used K
R_1, R_2	Random number
$H(\cdot)$	hash function
$\|$	connect
$\oplus$	xor

In the cloud-assisted BSN, there are three characteristics:

- The nodes are mobile, and they often move from one cluster to another.
- Each cluster has a head, which is the gateway node connected with the BSN cluster and the transmission subnet. The head has the non-limited communication band and is assumed to be secure as the traditional base station.
- Each cluster is like a traditional sensor network. The head has the same assumed abilities as the base station, and all the heads are connected with the transmission subnet and the cloud computing center.

We assume that the cluster has the security structures of the traditional WSN, such as SPINS [28], and the transmission subnet has the public key infrastructure just like the Internet. Here we focus on the authentication scheme for nodes' random cross-cluster roaming. Table 1 shows the notation used in the protocol.

3.1 Phase 1: Mobile Node Initial Registration

In the BSN, the mobile node M_A belongs to the home cluster A with a head CH_A (cluster head A), and registers in this local cluster. In the initial registration phase, M_A sends the registration request to CH_A. Then CH_A randomly selects a symmetric session key $K_{CH_A-M_A}$, a random number r and an identity authentication material $E_{sk_{CH_A}}(CH_A, M_A, t_b, t_e)$, where sk_{CH_A} is the private key of CH_A, t_b and t_e are the predefined beginning and ending time of the identity authentication, respectively. Thus, $t_e - t_b$ is the effective time of the identity authentication. As a reply, CH_A sends $K_{CH_A-M_A}$, r and

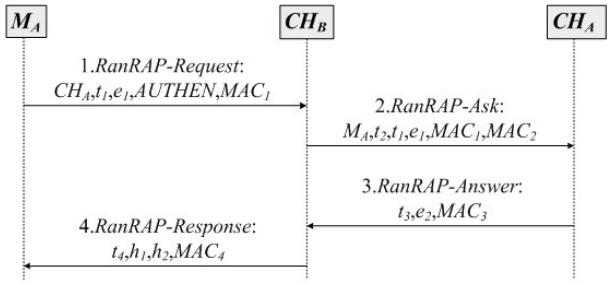


Figure 2: Random roaming authentication protocol

$E_{sk_{CH_A}}(CH_A, M_A, t_b, t_e)$ to M_A. M_A stores the information into its main memory. At the same time, CH_A saves the registration information. The initial registration is finished offline, and assumed to be secure.

3.2 Phase 2: Random Mobile Node Roaming Authentication Protocol (RanRAP)

In Phase 2, M_A moves to a new cluster CH_B, and acquires a legal identity in this foreign cluster. Besides, a new session key is generated between M_A and CH_B. The *RanRAP* protocol is described in Figure 2.

1) *RanRAP-Request phase.* M_A randomly selects a random number $R_1 \in \{0,1\}^a$, computes $e_1 = E_{K_{CH_A-M_A}}(R_1)$, $MAC_{K_{CH_A-M_A}}(CH_A,\ t_1,\ e_1,\ AUTHEN)$ and sends the *RanRAP-Request* message to CH_B,

$$M_A \longrightarrow CH_B : CH_A, t_1, e_1, AUTHEN, MAC_1 \quad (1)$$

where $AUTHEN = E_{sk_{CH_A}}(CH_A,\ M_A,\ t_b,\ t_e)$, $MAC_1 = MAC_{K_{CH_A-M_A}}(CH_A,\ t_1,\ e_1,\ AUTHEN)$, and t_1 is the presented time.

2) *RanRAP-Ask phase.* After receiving the *RanRAP-Request* message at time t^*, the foreign cluster CH_B checks whether $(t^* - t_1) \leq \Delta t$, Δt is a predefined threshold of time slot. If it is in a valid time interval, CH_B uses the public key $K_{PK_{CH_A}}$ of CH_A to decrypt AUTHEN and get CH_A^* and M_A^* in the cipher-text. The accuracy and the reliability of M_A are authenticated and some illegal messages are dropped. We can also obtain t_e which can resist the expired authentication non-limited reused by the adversary.

After the correctnesses of $CH_A^* = CH_A$ and $M_A^* = M_A$ are verified, CH_B sends the message *RanRAP-Ask* to CH_A and the home cluster of M_A,

$$CH_B \longrightarrow CH_A : M_A, t_2, t_1, e_1, MAC_1, MAC_2 \quad (2)$$

where t_2 is the message sending time, and $MAC_2 = MAC_{K_{CH_B-CH_A}}(M_A, t_2, t_1, e_1, MAC_1)$.

3) *RanRAP-Answer phase.* After receiving the message *RanRAP-Ask*, CH_A verifies the legitimacy by using the verification algorithm *VerifyA* shown in Figure 3. Therefore CH_A obtains R_1, and sends the *RanRAP-Answer* message to CH_B,

VerifyA algorithm

if $((t^*\text{-}t_2) \leqq \triangle t)$
  then compute $MAC_2^* = MAC_{K_{CH_A-CH_B}}(M_A, t_2, t_1, e_1, MAC_1)$;
  if $MAC_2^* = MAC_2$
    then CH_A find $AUTHRN$ and $K_{CH_A-M_A}$ by M_A;
      compute $MAC_1^* = MAC_{K_{CH_A-M_A}}(CH_A, t_1, e_1, ATUEHEN)$;
    if $MAC_1^* = MAC_1$
      then compute $R_1 = D_{K_{CH_A-M_A}}(e_1)$;
    end if
  end if
end if

Figure 3: *VerifyA* algorithm

$$CH_A \longrightarrow CH_B : t_3, e_2, MAC_3 \quad (3)$$

where $e_2 = E_{K_{CH_A-CH_B}}(r, R_1)$, and $MAC_3 = MAC_{K_{CH_A-CH_B}}(t_3, e_2)$.

4) *RanRAP-Response phase.* When CH_B receives the message *RanRAP-Answer*, the first step is to verify the correctness of t_3 and MAC_3. If the validation fails, the session ends. Otherwise r and R_1 are extracted, and a random number $R_2 \in \{0,1\}^a$ is chosen. The session key is computed according to Equation (4).

$$K_{CH_B-M_A} = H(R_1||R_2) \quad (4)$$

Finally, CH_B sends the following *RanRAP-Response* message to M_A,

$$CH_B \rightarrow M_A : t_4, h_1, h_2, MAC_4 \quad (5)$$

where $h_1 = r \oplus H(R_1), h_2 = H(r) \oplus R_2$, and $MAC_4 = MAC_{K_{CH_B-M_A}}(t_4, h_1, h_2)$.

5) M_A receives the *RanRAP-Response* message, and verifies whether t_4 is within the threshold time. If not, the session ends. Otherwise, $r* = h_1 \oplus H(R_1)$ is computed. If $r* = r$, CH_B identity is proved, then M_A computes $R_2 = H(r) \oplus h_2$. Hence M_A can obtain the new session key from Equation (4). Then M_A checks the correctness of MAC_4, if MAC_4 is correct, the authentication completes, and the new session key is valid.

After completing the authentication and generating the session key, CH_B immediately distributes a new identity authentication and r' to M_A, and informs CH_A to delete the identity authentication material r of M_A. Thus M_A becomes a member of CH_B, and can take CH_B as the home cluster and move to another new foreign cluster.

Taking into account the issues of traceability and tracking, when M_A joins CH_B, CH_B redistributes a new ID to the mobile node. We assume that each cluster head has 2^{16} IDs. When the mobile node obtains the trust of the

new cluster, the cluster head selects a new unused ID for the mobile node. In this way, we can prevent the outside nodes to track the trajectory of the mobile node. At the same time, in order to let the lawful authority trace the movement of the mobile node, each cluster head maintains a source ID table which is like Table 2. The table includes the ID of the previous home cluster head, the ID of the mobile node in previous home cluster, the redistribution ID of the mobile node in the new cluster, and the time taken to join the cluster.

3.3 Protocol Security Analysis

Proposition 1. RanRAP *satisfies the forward security.*

Based on the Table 2, even if the attackers acquire the session key $K_{CH_C-M_A}$ between the mobile node M_A and the cluster node CH_C, it is still difficult to derive the session key used before, i.e. $K_{CH_B-M_A}$ .The session key between M_A and CH_B is determined by two random numbers R_1 and R_2, and they are separately transmitted by the ciphertext e_1 in Equation (1) and the XOR value h_2 in Equation (5).

If the attackers want to obtain the plaintext R_1 from e_1, they must know the session key $K_{CH_C-M_A}$ between M_A and CH_A. However $K_{CH_C-M_A}$ is a preloaded value and is assumed to be completely secure. Thus it is impossible to obtain the value of R_1 in our *RanRAP*.

R_2 is also difficult to know because it is only used in $h_2 = H(r) \oplus R_2$. If the attackers want to deduce R_2 from h_2, they should know the hash value $H(r)$. However r is also a preloaded value and it is as secure as $K_{CH_C-M_A}$.

Even if the attackers acquire the current session key of M_A, they can not derive the previous session key of M_A without R_1 and R_2. Thus the protocol satisfies the forward security.

The forward security also provides a privacy protection for the roaming node. When the roaming node joins the new cluster, other nodes and the physical capture attackers do not know which cluster the roaming node comes from. However the cluster head that acts as the AP of the cluster knows the privacy, thus the roaming node can also be traced by the authorized assistance of the AP.

Proposition 2. RanRAP *obtains the local identity authentication.*

In the roaming protocol *RanRAP*, there is no pre-shared information between CH_B and M_A. However, the *RanRAP-Request* message in Equation (1) contains the identity authentication $AUTHEN$ which is encrypted by the private key of CH_A. After CH_B receives $AUTHEN$, CHB decrypts the ciphertext by using the public key of $K_{pk_{CH_A}}$ and obtains CH_A* and M_A*. If $CH_A* \neq CH_A$ and $M_A* \neq M_A$, the mobile node is judged to be illegitimate. The *RanRAP-Ask* message is not sent to CH_A in the BSN. In this way, the performance of the resistance against the forgery attack can be improved. CH_B can also acquire t_e, to resist the non-limitation of reusing the expired identity.

With the support of the MWN-based architecture, we assume that the authentication materials are securely transmitted by the transmission subnet. In the BSN, the authentication protocol serves as the local authentication schemes between the mobile node and the foreign header. That is to say, the MWN-based IoT architecture makes all the heads like the neighbors, which saves the computation and communication over the BSN.

Proposition 3. RanRAP *completes the mutual identity authentication.*

In the *RanRAP* protocol, M_A applies to join a new cluster by sending the *RanRAP-Request*. This message contains the authentication content $AUTHEN$. According to Proposition 2, $AUTHEN$ can achieve the initial identity authentication of M_A in the cluster CH_B. The completed identify authentication is realized by the algorithm *VerifyA* after CH_A receives the message *RanRAP-Ask* in Equation (2).

As CH_B shows his identity to M_A, it is mainly deterred by the random value r, which exists in the *RanRAP-Response* message. The random value is delivered through the XOR value $h_1.r$ is generated in the home cluster and is re-generated after the authentication is realized in the foreign cluster.

Proposition 4. RanRAP *has the ability of preventing man-in-the-middle attacks.*

According to the analysis of the *RanRAP* protocol, we find that the attackers are able to trick or truncate *RanRAP-Request* messages to imitate the mobile node M_A and communicate with the foreign cluster head. Thus the attackers can preserve the protocol and eventually be able to extract the session key material from the feedback message *RanRAP-Response*. However, according to Proposition 1, R_1 and R_2 are not sent in the plaintext. In order to attack the protocol, the previous session key should be known. The whole problem is back to the question of Proposition 1. For the intermediary tampering attacks, as mentioned in Proposition 3, the bilateral identity authentication ensures the correctness of the identity of the message sender, and the MAC used in every message ensures the message integrity. The security of the MAC depends on the security of the hash function. The recommended MAC size in our protocol is 4 bytes for the practical application, since only 40 forgery attempts per second are available in a 19.2 kb/s channel and 2^{31} trials are required for a successful forgery. The intermediary can not construct a valid message to realize the communication, thus the protocol is secure against the man-in-the-middle attacks.

Proposition 5. RanRAP *blocks the replay attack.*

The current timestamp is bound in every message of *RanRAP*. They are noted as t_1, t_2, t_3 and t_4 in Request, Ask, Answer and Response messages, respectively. If the received messages is not in the valid time slots Δt, it will

Table 2: Source ID table in the cluster head

Mobile Node	Source Cluster Head	Local ID	Time
M_A	CH_A	M'_A	t_a
M_B	CH_C	M'_B	t_b
...	...	...	...
M_I	CH_J	M'_I	t_k

be dropped to resist the replay attacks. The random number used to generate the pair-wise key is updated when the mobile node joins the new cluster according to Equation (5). Thus, the freshness and prevention of the replay attack are guaranteed validly.

3.4 Discussion about the DoS Exhaustion Attacks

The Denial of Service (DoS) attack is a key issue that must be considered in the design of the security network protocols. References [15] and [27] reported that the DoS attack can be efficiently prevented if the authentication is completed by the mobile node and the foreign head. That is to say, local roaming authentication at foreign head is beneficial for the DoS prevention.

Our *RanRAP* designs for the BSN based on the cloud-assisted structure (Figure 1). We assume the security questions on the transmission subnet are solved, and do not discuss the security question within a single BSN cluster with the assumption that is the same as traditional wireless sensor networks. Thus we consider the DoS attack on the *RanRAP* protocol, which is different from the DoS attack in WSNs discussed in [15, 27, 32]. The DoS attacks that we will study in the roaming authentication scenario can be classified into the following two aspects.

1) Attack from inside adversary. The inside attacks are launched by inside nodes. If the mobile node is physically captured by the adversary, it is compromised and replicated. Then a large number of replicas are deployed in the BSN, and the adversary can launch the DoS attack. Due to the fact that the multiple replicas have the same ID in the cluster, the cluster head is easy to find the replica by binding the sensor's relative location and ID. The replica detection is another research area and some papers have reported good results [35, 41]. When the replicas are deployed in different clusters, they are difficult to be detected by the ID recognition. However, in our *RanRAP*, the mobile node has a pair of keys with the cluster head. Only one replica is allowed in a cluster. Thus it is impossible for a simple replicated node to launch the DoS attack in a subnet. Thus the inside DoS attack is resisted by this means.

2) Attack from outside adversary. This kind of DoS attack often depletes the network resources by replaying the forged or overdue packets. *RanRAP* resists this attack by encrypting and authenticating the fresh message with a timestamp. Unfortunately, the cryptology algorithm can recognize the outside attacker, but can't fight against the resources depletion in communication and computation. The attackers can also cheat the sensor node by ceaselessly sending the request message to the header and asking for joining the cluster. Then the relay nodes forward a large amount of packets to the cluster head. The head runs *RanRAP* to authenticate the request.

Our *RanRAP* protocol can't prevent the outside DoS attack, because the sensor node directly sends the *RanRAP-Request* message to the cluster head. Because of the characteristic of the random roaming, the sensor node doesn't know any information about the mobile node when it joins the cluster. Thus it's difficult to authenticate the mobile node. The outside DoS attack is an open problem in this area, and Qiu's [29] and Han's [13] papers didn't consider the energy overhead caused by this attack.

In our cloud-assisted BSNs, *RanRAP* can be improved by dividing each BSN cluster into some small sub-clusters. This method was enlightened by the scalable and clustering scheme presented by Reference [20]. A sub-cluster can vote a chair by some cluster selection algorithms. When the mobile node M_A moves into the cluster CH_B, M_A first communicates with the closest sub-cluster chair CH_{B_i}. The validity of message *RanRAP-Request* is checked by CH_{B_i} as the first step. After check, the chair decides whether the request will be forwarded. In this improvement, we shift the verification process from CH_B to small cluster head CH_{B_i}. Then the bad consequence of the DoS attack is limited in a small sub-cluster.

4 Composable Security Based on Authentication Test

The authentication test was proposed by Guttam [5, 10]. The authentication test is based on the security protocol formalization of the strand spaces theory and the challenge-response mechanism. The instance of authentication test is constructed by a special form of the data transmission characterized with the unique source property. The special form of data transmission completes the proof of the authentication properties of the proto-

col through proving the existence of the general nodes. The composable authentication test was also proposed by Guttman [9] in 2009, and is used to prove that two protocols used in composition don't undermine the overall security.

4.1 Basic Framework of Composable Protocol's Authentication Test

The basic goal of the composable authentication test is to test whether Π_2 has a composable security. We consider the composition of protocols Π_1 and Π_2 (the composable protocol is denoted by Π_1). If the composable protocol Π is still able to achieve the security goals identified by II_1, it means the operation of Π_2 doesn't affect the security goals identified by Π_1. Thus Π_2 has a composable security based on the authentication test.

When the proposed protocol *RanRAP* runs, the cluster heads CH_A and CH_B have a shared key $K_{CH_A-CH_B}$. It can be assumed that $K_{CH_A-CH_B}$ is generated by the classic protocol TinyPK. Eventually, there are some circumstances of the composable using of TinyPK [37] and the *RanRAP* protocol. We record TinyPK as Π_1 and *RanRAP* as Π_2. Π means that TinyPK is used in the combination with the *RanRAP* protocol, which is used to test whether the *RanRAP* affects the security goals of TinyPK during the running process of Π. In other words, if Π achieves authentication test, then Π_1 is composable secure in this application instance.

The proof of the composable authentication test is generally executed in three steps [9]. First, the strand space directional figure is used to describe the initial protocol. It simplifies the running process of the protocol. And predicate symbol is also used in this protocol. Second, the security goal of Π_1 is deduced on the basis of the protocol logical description, and it is proved that the security goal of the composable protocol Π and Π_1 is homomorphic. At last, the node roles involved during the protocol running on the basis of the protocol logical description are defined and described. By decomposing the node role (Π_1 or Π_2) during the protocol execution of Π, the proof about the strong disjoint encryption of Π_1 and Π. the solution of counterexample of Π-skeleton, and the realization of Π-skeleton are completed. After the three steps, the security goals of Π_1 can be achieved, which means that the composable security authentication test of Π_2 is completed. The three steps are separately described in Subsections 4.2, 4.3, and 4.4.

4.2 Test Strand Space Model and Description of Protocol

As described in Subsection 4.1 the composable security of *RanRAP*(Π_2) is tested by the composable use states of TinyPK(Π_1) and *RanRAP*(Π_2) among nodes M_A, CH_A and CH_B. Thus during the composable protocol's execution, we can see whether the security goals of TinyPK(Π_1) can be realized. The strand space model is used here to simplify the two protocols, and the logical language is used to describe the protocols.

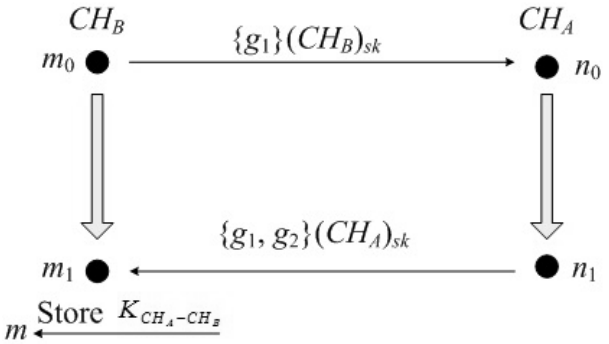


Figure 4: Strand space of TinyPk(Π_1)

The function of TinyPK [37] in the composable protocol is to generate the shared key among clusters. The strand space of TinyPk(Π_1) is shown in Figure 4.The cluster heads CH_B and CH_A are two participants. CH_B, m_0, and m_1 are the nodes of CH_B. n_0 and n_1 are the nods of CH_A. g_1 and g_2 are generated by the Diffie-Hellman key exchange. g_1 is g^x mod P, and g_2 is g^y mod P (x and y are random values). $\{g_1\}(CH_A)_{sk}$ represents that CH_A uses the private key of CH_A to encrypt g_1. Store $K_{CH_A-CH_B}$ means that after CH_B verifies the correctness of the messages sent by CH_A, CH_B computes the shared key $K_{CH_A-CH_B}$ and stores the key. For simplicity, some unnecessary parameters are neglected during the implementation of the protocol. The basic security assumptions of protocol TinyPK(Π_1) are as follows: (1) $(CH_B)_{sk}$, $(CH_A)_{sk} \notin K_P$ (K_P is the key set grasped by the penetrator), (2) x and y are generated uniquely, (3) g is not leaked, and (4) $x \neq y$.

As shown in Figure 4, the strand spaces of protocol Π_1 contain the initiator strands and the responder strands.

The initiator strands of protocol TinyPk(Π_1) are denoted by S_{i1}.which contains two participants CH_B and CH_A. Messages $\{\{g_1\}(CH_B)_{sk}\}$ and $\{\{g_1, g_2\}(CH_A)_{sk}\}$ are transmitted between them. We use $Init[]$ as the identity of the initiator strands

$$S_{i1} \in Init[CH_B, CH_A, \{g_1\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}].$$

The trace is described as Equation (6)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \; & < +\{\{g_1\}(CH_B)_{sk}\}, -\{\{g_1, g_2\}(CH_A)_{sk}\} > \end{aligned} \tag{6}$$

In Equation (6), $< \sigma_1, a_1 >$ represents the symbolic term of the trace. The symbol term generally are denoted by $< \sigma, a >$, where σ has the positive or negative values, corresponding to the sender or receiver, respectively. a is the strand space trajectory, which is the message transmission path. For example, the trace of the initiator $< \sigma_1, a_1 >$ corresponding to $+\{\{g_1\}(CH_B)_{sk}\}$, means that CH_B sends a message $\{g_1\}(CH_B)_{sk}$.

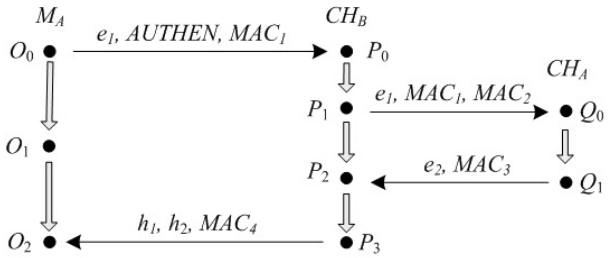


Figure 5: Strand space of TinyPk(Π_2)

∀ $m, x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$
if Store1(m, $(CH_A, CH_B)_K$)
∧Non(pk(CH_A)) ∧Non(pk(CH_B)) ∧Non(g) ($\varphi 1$)
∧Non(sk(CH_A)) ∧Non(sk(CH_B)) ∧Non(g) ($\varphi 2$)
then ∃ m_0, n_0
$CH_B1(m_0, x, g_1, .(CH_B)_{sk}) \wedge CH_A1(n_0, g_1, (CH_B)_{pk})$
and ∃ m_1, n_0
$CH_A2(n_1, g_1, y, g_2, (CH_B)_{pk}, (CH_A)_{sk}) \wedge CH_B2(m_1, x, g_1, g_2, (CH_B)_{sk}, (CH_A)_{pk})$
∀ $m, n, x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$
if $\varphi 1 \wedge \varphi 2 \wedge$ Lsn1(n, g)
then false

Figure 6: Implementation of Protocol TinyPK(Π_1)

The responder strands of protocol TinyPk(Π_1) are denoted by S_{r1} which contains the same strands as the initiator strands. We used $Resp[]$ as the identity of the responder strands, then

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \ & < -\{\{g_1\}(CH_B)_{sk}\}, +\{\{g_1, g_2\}(CH_A)_{sk}\} > \end{aligned} \tag{7}$$

$< \sigma_1, a_1 >$ corresponds to $-\{\{g_1\}(CH_B)_{sk}\}$ in Equation (7), means that CH_A receives a message $\{g_1\}(CH_B)_{sk}$. $< \sigma_2, a_2 >$ in Equation (7) corresponds to $+\{\{g_1, g_2\}(CH_B)_{sk}\}$, and means that CH_A sends a message $\{g_1, g_2\}(CH_A)_{sk}$.

The function of protocol *RanRAP* in the composable protocol is to achieve the mobile node authentication accessing to the new foreign cluster by using the shared key generated by protocol TinyPK. The strand spaces are shown in Figure 5, where O_0, O_1 and O_2 are the nodes of participant M_A, P_0, P_1. P_2 and P_3 are the nodes of participant CH_B. Q_0 and Q_1 are the nodes of participant CH_A. The symbols involved in strand space have the same definition as described in Subsection 3.2. The basic security assumptions of protocol *RanRAP*(Π_2) are as follows: (1) $(M_A, CH_A)_k$, $(CH_A, CH_B)_k \notin K_P$, (2) R_1 and R_2 are generated uniquly, and (3) $R_1 \neq R_2$. $(M_A, CH_A)_k$ denotes the session key between M_A and CH_A, and K_P is the key set grasped by the penetrator.

In Figure 5, the basic strand spaces of protocol Π_2 contain the initiator strands, the responder strands, and the server strands.

The initiator strand of protocol *RanRAP*(Π_2) is denoted by S_{i2}, which contains two participants, M_A and CH_B. The messages transmitted between them are $e_1, AUTHEN, MAC_1$ and h_1, h_2, MAC_4. We use $Init[]$ as the identity of the initiator strands.

$$S_{i2} \in Init[M_A, CH_B, e_1, AUTHEN, MAC_1, h_1, h_2, MAC_4].$$

The trace is described as Equation (8)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \ & < +\{e_1, AUTHEN, MAC_1\}, -\{h_1, h_2, MAC_4\} > \end{aligned} \tag{8}$$

The responder strands of protocol *RanRAP*(Π_2) is denoted by S_{r2}, which contains three participants, M_A, CH_B and CH_A. The messages transmitted among them are $\{e_1, AUTHEN, MAC_1\}$, $\{e_1, MAC_1, MAC_2\}$, $\{e_2, MAC_3\}$ and $\{h_1, h_2, MAC_4\}$. We use $Resp[]$ as the identity of the responder strands, then

$$S_{r2} \in Resp[M_A, CH_B, CH_A, e_1, AUTHEN, MAC_1, MAC_2, e_2, MAC_3, h_1, h_2, MAC_4].$$

The trace is described as Equation (9)

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >, < \sigma_3, a_3 >, < \sigma_4, a_4 >> \\ = \ & < -\{e_1, AUTHEN, MAC_1\}, \\ & \quad +\{e_1, MAC_1, MAC_2\}, -\{e_2, MAC_3\}, \\ & \quad +\{h_1, h_2, MAC_4\} > \end{aligned} \tag{9}$$

The server strands of protocol *RanRAP*(Π_2) is denoted by S_{s2}, which contains two participants, CH_A and CH_B. The messages transmitted between them are $\{e_1, MAC_1, MAC_2\}$ and $\{e_2, MAC_3\}$. We use $Ser[]$ as the identity of the server strands.

$$S_{s2} \in S_{er}[CH_A, CH_B, e_1, MAC_1, MAC_2, e_2, MAC_3].$$

The trace is described as Equation (10),

$$\begin{aligned} & << \sigma_1, a_1 >, < \sigma_2, a_2 >> \\ = \ & < -\{e_1, MAC_1, MAC_2\}, +\{e_2, MAC_3\} > \end{aligned} \tag{10}$$

4.3 Security Goals Description of Protocol Π_1 and Homomorphism Security Goals Proof of Π

Guttman [9] defined a goal language $L(\Pi)$ based on the classical first order logic for the strand space security protocol. $L(\Pi)$ is a language for the execution of protocol Π, based on Classic Quantified Language. We use $L(\Pi)$ to describe the execution of protocol Π and define the security goals of the protocol. Figure 6 shows the implementation of protocol TinyPK(Π_1) based on the language grammar, in which the strand space is described in Figure 4. φ_1 and φ_2 are extracted as the security goals.

In Figure 6, m is the storage node where CH_B stores $K_{CH_A - CH_B}$. m_0, m_1, and n_0 correspond to the nodes as shown in Figure 3, which send or receive a message. n is defined as the attackers' monitoring node which is not drawn out in Figure 4. $x, y, g, (CH_A)_{pk}, (CH_B)_{pk}, (CH_B)_{sk}, (CH_A)_{sk}$ and other notations in Figure 6 have the same definitions as Subsection 4.2 and Figure 4.

Figure 6 also includes some new predicate symbols, such as Store1 $(m,(CH_A,CH_B)_K)$ and Non$(pk(CH_A))$, which are defined as follows.

In the predicate symbols like Nov(v), v is assumed to be no-originating. It exists in every entity before the implementation of the protocol and is not grasped by the penetrator. Be specific to Non$(pk(CH_A))$, Non$(pk(CH_B))$, Non$(sk(CH_A))$ and Non$(sk(CH_B))$ in Figure 6, where $(sk(CH_B))$ is defined as the private key of CH_B, and $(pk(CH_B))$ is the public key of CH_B.

Predicate symbols like RhoJ$(m,v_1,\cdots,v_i)$ are the role predicate. They are defined as follows, In skeleton A when m is the j^{th} node of an instance of the role ρ, with its parameters (in some conventional order) instantiated by the associated values $v_1,\cdots,v_i$. Be specific to Store1$(m,(CH_A,CH_B)_K)$, $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$, $CH_A1(n_0,\ g_1,\ (CH_B)_{pk})$, $CH_{A2}(n_1,\ g_1,\ y,\ g_2,\ (CH_B)_{pk},\ (CH_A)_{sk})$, $CH_{B2}(m_1,\ x,\ g_1,\ g_2,\ (CH_B)_{sk},\ (CH_A)_{pk})$ in Figure 6, where Store1$(m,\ (CH_A,CH_B)_K)$ means that Store role stores $(CH_A,CH_B)_K$ in node m. $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$ means that CH_B produces variable x, g_1, $(CH_B)_{sk}$ at its first node m_0. The same explanation can be used in the other role predicates.

The security goals based on predicates φ_1 and φ_2 can be explained as follows. Before $K_{CH_A-CH_B}$ is generated, CH_A receives the message $\{g_1\}(CH_B)_{sk}$ and successfully obtains g_1. CH_B receives the message $\{g_1,g_2\}(CH_A)_{sk}$ and successfully obtains g_2. Security assumptions: $(CH_B)_{sk},(CH_A)_{sk}$ and g are not leaked.

The mapping relation is based on the strand space and classic quantity language. We can verify the security goals claimed by φ_1 and φ_2 in Figure 6 and the security goals in Figure 4 are homomorphic.

Theorem 1. ***The security goals based on strand space model in Figure 4 can be expressed as*** φ_1 ***and*** φ_2.

Proof. Suppose A as the skeleton of Π. σ is the function mapping of skeleton A from variables of Π_1 to the strand space of Π. To deduce $A \models \varphi(\varphi_1 and \varphi_2)$, all the function mapping of A should satisfy $A,\sigma \models \varphi$.

The propositional logic is defined via the standard Tarski inductive clauses for the classical first order logical constants, and the basic clauses are as follows:

- $A,\sigma \models Nov(v)$, iff $\sigma(v) \in non_A$ (non_A means passive, and it exists in the role node before the implementation of the protocol);
- $A,\sigma \models RhoJ(m,v_1,\cdots,v_k)$, iff $\sigma(m) \in nodes(A)$ (nodes(A) represents the nodes belonging to skeleton A), and $\sigma(m)$ is the j^{th} node of the role ρ with its parameters (in some conventional order) instantiated by the associated values $\sigma(v_1),\cdots,\sigma(v_k)$.

The predicates $pk(CH_A)$, $pk(CH_B)$, $sk(CH_A)$, and $sk(CH_B)$ in Figure 6 are no-originating in skeleton A. All the four predicates meet $\sigma(v) \in non_A$, and then $A,\sigma \models Nov(v)$.

Same as the role nodes in Figure 6, Store1(m, $(CH_A,\ CH_B)_K)$, $CH_B1(m_0,\ x,\ g_1,\ (CH_B)_{sk})$, $CH_A1(n_0,\ g_1,\ (CH_B)_{pk})$, $CH_{A2}(n_1,\ g_1,\ y,\ g_2,\ (CH_B)_{pk},\ (CH_A)_{sk})$, $CH_{B2}(m_1,\ x,\ g_1,\ g_2,\ (CH_B)_{sk},\ (CH_A)_{pk})$ are the role nodes and meets the parameter relationship, which satisfies $A,\sigma \models RhoJ(m,v_1,\cdots,v_k)$.

We define $A \models \varphi$ when $A,\sigma \models \varphi$ for all σ. Theorem 1 verifies that the security goals based on the strand space model in Figure 4 can be expressed as φ_1 and φ_2. □

4.4 Composable Security Proof of the RanRAP Protocol

According to the definition of the composable security protocol in [9], the realization is divided into the following steps. First, it is proved that Π and TinyPK(Π_1) have a strong disjoint encryption, which is defined here as Proposition 6. Second, we give the solution to the counterexample of Π_1-skeleton of Π and the realization of Π-skeleton as Proposition 7. Finally, after drawing the above two conclusions, with the composable security definition described in [9], Theorem 2 is concluded, which means that RanRAP is a composable security.

Proposition 6. *Protocol* Π *and TinyPK(*Π_1*) satisfy the Strong Disjoint Encryption.*

The strong disjoint encryption requires that when *RanRAP*(Π_2) constructs the protocol, there should be no creation conflicts and extraction conflicts with TinyPK(Π_1). The creation conflicts mean that *RanRAP*(Π_2) can not create encryptions which are specified in TinyPK(Π_1). *RanRAP*(Π_2) can be used, but can not construct a similar encryption, which can leak the contents constructed by TinyPK(Π_1). The extraction conflicts mean that the encrypted content of TinyPK(Π_1) are transmitted to the *RanRAP*(Π_2) protocol, which can not be re-transmitted the plaintext outside these encrypted contents again.

Proof. According to the definition of the strong disjoint encryption in [9], the proof is divided into three steps. First, the primary and secondary nodes of the combination protocol Π should be found out. Second, the message related to the creation conflicts and extraction conflicts should be found out, which is based on the definition of the secondary nodes. Finally, combined with the secondary node, the content encrypted by the creation conflicts and extraction conflicts should be found out, and the conclusion should be drawn. □

1) Determining the primary and secondary nodes. The primary nodes are defined as the role nodes, appeared in TinyPK(Π_1) when using the composable protocol Π. The role nodes are defined as the secondary nodes, which are used in protocol Π but not in an instance of the role nodes of TinyPK(Π_1). According to the definition of the role nodes and traces in Equations (6) and (9), it can be found that CH_A and CH_B are the primary nodes, and M_A is the secondary node.

2) Determining the contents of the creation conflicts and extraction conflicts about the secondary node. The main purpose of this step is to identify all the involved encrypted and decrypted contents of the secondary node, which is prepared for the creation conflicts and extraction conflicts of the next step. According to trace in Equation (8), the encrypted contents are e_1, $MAC_1$and $AUTHEN$, and

$$\begin{aligned} &e_1 \in E \subseteq (\Pi_2) = \\ &\quad \{e_1 : \exists O_0, \alpha. e_1 \subseteq msg(\alpha(O_0)) \wedge (O_0) \\ &\quad is\ a\ role\ node\ of\ \Pi_2\}, \end{aligned} \tag{11}$$

$$\begin{aligned} &MAC_1 \in E \subseteq (\Pi_2) = \\ &\{MAC_1 : \exists O_0, \alpha. MAC_1 \subseteq msg(\alpha(O_0)) \wedge (O_0) \\ &\quad is\ a\ role\ node\ of\ \Pi_2\}, \end{aligned} \tag{12}$$

$$\begin{aligned} &AUTHEN \in E \subseteq (\Pi_2) = \\ &\quad \{e_AUTHEN : \exists O_0, \alpha. \\ &\quad e_{AUTHEN} \subseteq msg(\alpha(O_0)) \wedge (O_0) \\ &\quad is\ a\ role\ node\ of\ \Pi_2\}. \end{aligned} \tag{13}$$

From Equations (11), (12), and (13), it can be known that the encrypted contents e_1, MAC_1 and $AUTHEN$ are implemented in protocol Π_2. In the strand space model, there is a corresponding homomorphism at node O_0 that generates the encrypted content, and O_0 is the role node of protocol Π_2.

3) Determining the strong disjoint encryption. The strong disjoint encryption requires the secondary node having no creation conflicts or extraction conflicts with the TinyPK(Π_1) protocol. According to Equation (11), e_1 has the creation encryption related with protocol *RanRAP*(Π_2) referred to the specific encryption content of e_1. It is not relevant to TinyPK(Π_1), and there is no creation conflicts. MAC_1 is the same as e_1. But for $AUTHEN$, it does not belong to TinyPK(Π_1) or RanRAP(Π_2), and the decrypted contents of $AUTHEN$ do not flow in the trace, which is only used as the middle validation. Even if the decrypted contents combined with the message which generates $AUTHEN$, there are no creation conflicts or extraction conflicts.

From the above analysis, we can find that the secondary node of the combination protocol Π does not cause the creation conflicts or extraction conflicts. Therefore Π and TinyPK(Π_1) satisfy the strong disjoint encryption.

Proposition 7. *The cluster counterexample of A of protocol* Π *in protocol* Π_1*(TinyPK) and the realization proof of cluster A in protocol* Π*.*

For any goal $G_1 \in L(\Pi_1)$, the TinyPK(Π_1)-counterexample A_1 from a Π-counterexample B should be squeezed. This can be achieved by the following two steps. First, B is restricted to its primary node $B \uparrow \Pi_1$(represented by cluster A). Then, all the non-primary encryptions $e \notin E \ll (\Pi_1)$ are removed from A, thus the rest is A_1.

B is first restricted to its primary node skeleton $B \uparrow \Pi_1$ form traces in Equations (6)-(10): $[CH_B, CH_A, \{g_1\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}, e_1, MAC_1, MAC_2, e_2, MAC_3]$.

After all the non-primary encryptions $e \notin E \ll (\Pi_1)$ are removed from A, skeleton A_1 is $[CH_B, CH_A, \{g^x\}(CH_B)_{sk}, \{g_1, g_2\}(CH_A)_{sk}]$.

Proof. The realization of skeleton A is achieved through the authentication test in [9]. There is a new proposed authentication test. Thus we first describe the definition of the authentication test as Lemma 1. □

Lemma 1. *Let c be an atom or an encryption, and S be a set of encryptions. If* $\exists n \subseteq nodes(A)$*, Cut*$(c, S, A)$*, is the test cut for c and S in A, we formalize*

$$\begin{aligned} &Cut(c, S, A) \\ = \ &\{n \subseteq nodes(A) : \exists m. m \leq_A n \wedge c \dagger^S msg(m)\}. \end{aligned} \tag{14}$$

According to the new definition of the authentication test in Lemma 1, two important cuts $\mathrm{Cut}(g_1, S_1, A)$ and $\mathrm{Cut}(g_2, S_2, A)$ should be solved in skeleton A. $\mathrm{Cut}(g_1, \{g_1, g_2\}(CH_A)_{sk}, A)$ is solved at node n1, and $\mathrm{Cut}(g_2, \{g_1, g_2\}(CH_A)_{sk}, A)$ is solved at node m_1. Thus skeleton A is realized.

The final judge of the composable protocol Π_2 is based on the composable theorem in [9]. We describe it as Lemma 2 here.

Lemma 2. *Let* Π *and* Π_1 *have the strong disjoint encryption, and let* $G_1 \in L(\Pi_1)$ *be a security goal. If* $A \models\!\rightharpoonup G_1$ *can be realized, for some realized* Π_1*-skeleton* A_1*,* $A_1| =\!\rightharpoonup G_1$*.*

Theorem 2. ***RanRAP is a composable security****.*

Proof. In Theorem 1, $A \models \varphi$ (φ is expressed as two secure claims, φ_1 and φ_2), and Proposition 7 has proved that the skeleton A of protocol Π is realized. Combined with the definition of the counterexample realized in [9], the conclusion $A| =\!\rightharpoonup G_1$ is drawn.

According to Lemma 2, the first requirement of protocols Π and Π_1 is that they should have a strong disjoint encryption, which has been described in detail in Proposition 6. Another requirement of Lemma 2 is that $A| =\!\rightharpoonup G_1$ should be met, which has also been deduced from Theorem 1 and Proposition 7. Hence Π and Π_1 also satisfy another premise of Lemma 2. A_1 has been given as the counterexample of Π_1 in Π. According to Lemma 2, the conclude $A_1| =\!\rightharpoonup G_1$ is drawn. The composable protocol Π does not affect the security goals of protocol TinyPK(Π_1). Thus the composability of the *RanRAP* protocol is concluded. □

5 Protocol Performance Analysis and Comparison

The roaming authentication protocol of the cloud assisted BSN has three aspects of security needs: lightweight, random roaming and composable security. In this section, our work is compared with the related work in terms of these three aspects both qualitatively and quantitatively.

5.1 Comparison with the Related Works

Table 3 lists the comparison between *RanRAP* and related protocols in the aspects of lightweight, randomly roaming and security. The computation overhead is measured by CPU's number of revolutions in a 8 MHz CPU. The message size is measured in the unit of byte. In the following, h_n is the average hops when the mobile node in a cluster reaches the cluster head, n_c is the average number of the neighboring clusters, and " $-$ " means that it is not considered in the security part.

The lightweight is compared in three aspects: communication times, computation overhead and message size. In the comparison of communication times, Han 2010 [13] and our *RanRAP* protocol consider the entire authentication process. The communication times include the transmission time of the authentication materials. The roaming protocols of Yang 2010 [39] and He 2011 [12] 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 former total communication times are greater than communication times involved in the mobile nodes. In the roaming protocol, the mobile node is limited by the resource. Thus the lightweight focuses on mobile nodes. The sensor node in IoT under the mobile environment is more limited in energy, computation capacity and communication capacity. Thus it has more demands on the lightweight. The communication times of Han 2010 [13] are equal or greater than 4 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.

The numbers of CPU revolutions are used to calculate the computation overhead. It is mainly based on [26] which proposed to use the energy consumption relationship of each algorithm to estimate the results. In the 8 MHz CPU for the Micaz mote, its encryption algorithm, CPU revolution and energy consumption are shown in Table 4.

Table 5 shows the basic cryptographic operations used in the roaming protocol. The cryptographic algorithms corresponding to the energy consumption is shown in Table 4. BCE represents Block Cipher Encryption, *MAC* means Message Authentication Code Computer, PKE means Public Key Encryption or Decryption, ECSM means Elliptic Curve Scalar Multiplication, P means El-

Table 3: Comparison of related work

protocol	lightweight						Random roaming	security		
	Communication-ation times		Communication overhead (CPU revolution)		Message size			Local identity authentication	Prevent Dos attack	Composable security
	Whole protocol	Mobile node	Whole protocol	Mobile node	Whole protocol	Mobile node				
Yang 2010[9]	3	3	393.225M	198.45M	$74h_n + 72$	$74h_n + 72$	$\surd$	-	$\surd$	-
He 2011[10]	3	3	584.225M	294M	$188h_n + 72$	$188h_n + 72$	$\surd$	-	$\surd$	-
Han 2010[21]	≥ 4	3	150.388K	53.71K	$52h_n + 88$ $+56n_c$	$52h_n + 88$	$\times$	-	$\surd$	-
Our *RanRAP*	4	2	25.529M	32.226K	$50h_n + 108$	$50h_n + 28$	$\surd$	$\surd$	$\surd$	$\surd$

Table 4: Energy algorithm consumption

Encryption algorithm	Energy consumption	CPU revolution
$AES(128bits)$	$38\mu J$	10742
$ECDSA(160bits)$	$52mJ$	$14.7M$

liptic Curve Bilinear Pairing, and EXP means Modular Exponentiation. The comparison of the computation overhead does not consider the hash algorithm overhead when the protocols run.

Table 5: Protocol encryption operations

Protocol operation	Energy consumption
BCE	$1AES$
MAC	$1AES$
$ECSM$	$1ECDSA$
P	$6ECDSA$
PKE	$2ECDSA$
EXP	$2ECDSA$

The basic computation times of the protocols in Table 3 are shown in Table 6. From Tables 4, 5 and 6, the estimated calculation of the computation overhead can be obtained.

Table 6: The whole protocol computation overhead

Protocols	Whole computation overhead	Whole energy overhead
Yang 2010 [39]	$8.75ECSM + 3P$	$1391mJ$
He 2011 [12]	$15.75ECSM + 4P$	$2067mJ$
Han 2010 [13]	$4BCE + 8MAC$	$456\mu J$

From Table 3, we can find that computation overhead of Han 2010 [13] and *RanRAP* are lower than the similar protocols in an integer magnitude on the mobile sensor nodes. This is mainly determined by the goal of the protocol design. We can obtain the computation time of each protocol based on the CPU revolutions from Table 3. The computation times of Yang 2010 [39] and He 2011 [12] are 24.08625 s and 36.75 s in the 8 MHz CPU for the Micaz nodes, respectively. The theoretical value of computation times does not include the time overhead of the communication delays and the task waiting, but it is still too long for the roaming service. While Han 2010 [13] and *RanRAP* are 6.7125 ms and 4.0256 ms, respectively, it has an obvious applicability regarding the computation overhead. In the comparison of the lightweight between Han 2010 and *RanRAP*, *RanRAP* is more excellent than Han 2010, because *RanRAP* uses fewer cryptographic operations. This is mainly due to the reduced communication times while the average amounts of the computation time of the two protocols are almost the same. In addition, it should be noted that when the local authentication is activated, the number of the consumed revolutions is 25.529 M which is much bigger than Han's solution. However, the decryption algorithm of the public keys runs on the cluster head, which has enough energy to support. If the function of the local authentication is inactivated, the whole process costs 128.904 K.

Table 3 shows the comparison of the message size. We consider the influence of the average relay hops in the cluster and the average adjacent clusters on the entire message size of the protocol (h_n means the average relay hops in the cluster, and nc means the average neighboring clusters). The message size in Table 3 is measured by the data in Table 7. In addition, in the message size calculation process, the message format is beyond calculation, and the message size of the symmetric encryption is an integer multiple of the key length. The message size of the public encryption is only calculated as the size of the encrypted message. The length of the hash value has the same length as the content in the hash function.

From the message size in Table 3, we can find that the message sizes of Han 2010 and *RanRAP* protocol are the shortest. Despite using the public key algorithms, the length of the protocol message only computes according to the length of the encrypted content without considering the specific public key algorithm. However, with the number of relay hops in the cluster increased, the *RanRAP* protocol has a better performance. In addition, Han 2010 protocol is also related with the average number of the adjacent cluster n_c. When n_c is large, the protocol roaming range and the message size are enlarged. Other protocols can execute the random roaming without relying on the location of the home cluster and foreign cluster.

RanRAP protocol uses $AUTHEN$ to filter some illegal joiners by decrypting parameters in the local foreign cluster before the mobile node is authenticated. The specific instructions can refer to the analysis of Proposition 2 in Subsection 3.3.

For the security comparison, some roaming protocols should go to the fixed home cluster to achieve the identity authentication. Thus the DOS attack is unavoidable. In Han 2010 and *RanRAP* protocols, when a mobile node reaches a new cluster, the mobile node doesn't need to go back to the fixed initial cluster to obtain the authentication material in the next roaming, which reduces the harm caused by the DOS attacks. What's more, we propose

Table 7: Bytes of basic protocol

Protocol content	bytes
MAC	4
Time stamp	8
Random number	8
Key	16
Node Id	2
Index operation	50
Elliptic curve length	20

a solution to the DOS attacks caused by the Multi-hop transmission. The details are shown in Subsection 3.4. The composable security is the most distinctive feature of the *RanRAP* protocol compared with other lightweight protocols. That is because when the lightweight protocol is used by the cluster head, the cluster head acts as the gateway which connects the BSN subnets and the backbone network. The lightweight protocol running in the BSN cluster can not affect the security goals of the original protocol. Section 4 completes the composable security proof of the authentication test.

5.2 Protocol Simulation

We simulate the *RanRAP* protocol by using NS2.29, and the transmission delay is used to quantify the message sizes, because the transmission delay can reveal the availability and efficiency of the *RanRAP* protocol. The simulation uses the mesh network topology. The MAC layer uses the 802.15.4 protocol which Zheng [40]wrote for NS2. The routing layer uses the AODV routing protocol which has the shortest hops. The transportation layer uses the UDP protocol, and the application layer transmits the CBR packet. The message size is set according to Table 7 in Subsection 5.1. The data transmission speed is 250 KB/s, which adopts the recommended beacon mode standard setting in [40].

Supposing the communication radius of the mobile node and the sensor nodes within the cluster is 20 m, the communication radius of the cluster head is 100 m. The delay times of the *RanRAP-Request* and *RanRAP-Response* messages in the node are derived from the computation overhead of the mobile node according to Table 3 (using the Micaz mote with 8 MHz CPU). The delay times are 3 ms and 1.5ms. The time delay of *RanRAP-Ask* and *RanRAP-Answer* in the node is set to 1 ms (using a CPU with a frequency of a few hundreds MHz as the cluster head processor).

To analysis the message size of Table 3 in 6, we design two groups of simulations. The number of each simulation is 100 times.

Simulation 1: When $n_c = 1$, we set $h_n = 2$ and $h_n = 5$. When the protocol runs, only a specified neighboring

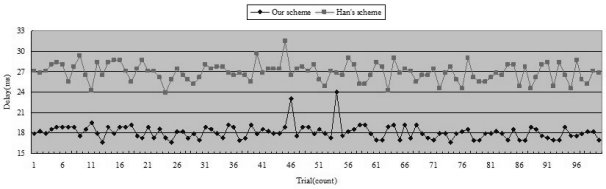


Figure 7: Time delay for $n_c = 1$ and $h_n = 2$

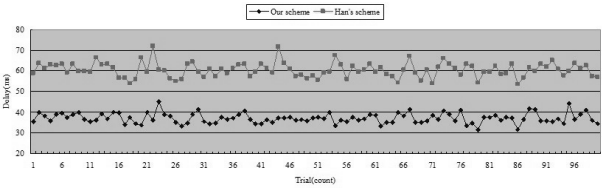


Figure 8: Time delay for $n_c = 1$ and $h_n = 5$

cluster can roam, and *RanRAP* degenerates as the Han 2010 [13] protocol which is a re-authentication protocol between clusters. Figures 7 and 8 show 100 sets of data obtained from the simulation.

In Figure 7, $n_c = 1$, and $h_n = 2$. The average delay of Han 2010 is 26.864 ms, whereas the average delay of *RanRAP* is 18.1056ms. Contrary to the theoretical analysis of the message size, the delay of the *RanRAP* simulation is less. This is mainly due to the *RanRAP* protocol less once to send a message. When *RanRAP* sends a message, it needs to add the MAC layer header, which makes the simulation delay is not proportional to the message sizes. The fluctuation effect in Figure 7 is mainly caused by $h_n = 2$. During the transmission process of nodes in the cluster, it needs to consume the transmission delay (when the node transmits, it needs to repeat calling, sending and receiving process, and seek the routing table, which needs more delay time). Thus the time is unstable.

In Figure 8, $n_c = 1$ and $h_n = 5$. The average delay of Han 2010 is 60.3689 ms, whereas *RanRAP* is 37.04ms. With the relay hops in the cluster increased, the advantages of the *RanRAP* protocol are more obvious compared with the results in Figure 7. This is due to less communication message sizes of the mobile node. From the comparison between Figures 7 and 8, we can also see that the volatility becomes larger, which is due to $h_n = 5$. This indicates that as the cluster relay hops increase, the instability of the delay is more obvious. Table 3 also shows that the *RanRAP* protocol has the characteristic of the random roaming. In order to reflect the performance advantages by using the network model, we design simulation 2. The simulation assumes that the mobile node randomly roams in a fixed region which has 100 clusters. The average number of the neighboring clusters around each cluster head is $n_c = 4$. In order to reflect the fairness, we assume that the Han 2010 protocol can

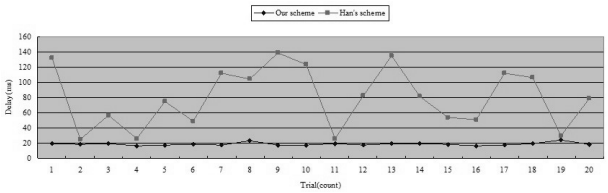


Figure 9: Time delay for $n_c = 4$ and $h_n = 2$

automatically search for the shortest hop to roam to the foreign cluster.

Simulation 2: When $n_c = 4$ and $h_n = 2$, the mobile node roams randomly in the region, and the mobile nodes separately run the *RanRAP* and Han 2010 protocols. We randomly select the foreign cluster in the simulation. The mobile node which runs the *RanRAP* protocol only needs to complete the authentication protocol once, whereas the Han 2010 protocol requires to inquiry the shortest path and then authenticates one cluster by one cluster. Figure 9 shows 20 clusters which are randomly selected to visit, and their simulation times are calculated by selecting the foreign cluster to joining the visit cluster (The maximum roaming hop is 5 and the minimum is 1.).

Figure 9 shows the experimental results of 20 times. The abscissa shows that the crossing cluster roaming hop of the 2nd, 4th, 11th, and 19th experiments is 1. The two protocols have a similar performance in the experiment. The hop of the 3rd, 6th, 15th, and 16th experiments is 2. The hop of the 5th, 12th, 14th, and 20th experiments is 3. The hop of the 7th, 8th, 17th, and 18th experiments is 4. The hop of the 1st, 9th, 10th, and 13th experiments is 5. Under these circumstances, the time delays of Han 2010 are 2, 3, 4, and 5 times that of the *RanRAP* protocol, respectively, because Han 2010 needs to join the neighboring cluster to authenticate several times through the running path.

Note that the message sizes of Han 2010 are proportional to the number of n_c. However, in simulation 2 only one message transmission delay is included. We do not consider the cluster head waiting for the message to send by sequence. In addition, the fluctuation of the same roaming hop is in a wide range as shown in Figure 9. As the hops among the clusters increases during roaming in Han 2010, the direct ratio of hops to the time delays is not obvious. This is due to the accumulation of the transmission fluctuation.

We further demonstrate the random roaming characteristics of the *RanRAP* protocol in Figure 9. It also illustrates the application network model considered by Han 2010, which restricts its advantages in terms of roaming. From the above time delay measured by the simulation, we can find that the whole time delay of the *RanRAP* protocol can be limited within 50 ms. Within the tolerance for the time delay roaming protocol in [22], the normal use and the normal operation of the node itself is not affected, which can achieves a seamless interface in the practical application.

6 Conclusion

With the application and development of the BSN, the BSNs become popular and are distributed widely. Then many BSN clusters are connected with the backbone transmission networks, and the big data collected by BSN require cloud storage and processing. Thus a novel type of cloud-assisted BSNs is presented. We consider the security questions of this type of BSNs with cloud-assisted infrastructure. Especially, we discussed the roaming authentication of the mobile body sensors in this scenario. In this paper, we exploit the advantages of cloud-assisted BSNs based on MWN model, and design an efficient, secure and composable protocol for the mobile nodes roaming randomly in the networks. The security analysis shows that our designed protocol can satisfy the forward security and mutual identity authentication, and can prevent the man-in-the-middle attacks and the replay attacks. The performance analysis shows that the *RanRAP* protocol can achieve lightweight, random roaming and composable security, which is well adapted to the application requirements of the BSN based on cloud-assisted infrastructure.

Acknowledgments

The paper has been supported by Natural Science Foundation of China under No.61003300 and 61272074, and Natural Science Foundation of Jiangsu Province under No.BK2011464. We would like to acknowledge that Prof. QIU Ying and Robert DENG Huijie greatly improved our work and advised us to discuss DoS attack prevention.

References

[1] M. Burmester, T. V. Le, B. D. Medeiros, and G. Tsudik, "Universally composable RFID identification and authentication protocols," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 4, pp. 21-33, 2009.

[2] C. C. Chang and H. C. Tsai, "An anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Transactions on Wireless Communication*, vol. 9, no. 11, pp. 3346–3353, 2010.

[3] S. Chari, C. Jutla, and A. Roy, "Universally composable security analysis of Oauth v2.0," *IACR Cryptology ePrint Archive*, pp. 526, 2011.

[4] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V.Leung, "Body area networks: A survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[5] S. F. Doghmi, J. D. Guttman, and F. J. Thayer, "Completeness of the authentication test," in *The 12th European Symposium on Research in Computer Security (ESORICS 2007)*, LNCS 4734, pp. 106–121, Springer, 2007.

[6] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," in *The 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.58–67, Chicago, IL, 2005.

[7] T. Feng, W. Zhou, and X. Li, "Anonymous identity authentication scheme in wireless roaming communication," in *2012 8th International Conference on Computing Technology and Information Management (ICCM'12)*, vol. 1, pp.124–129, Seoul, Korea, 2012.

[8] G. Fortino, M. Pathan, and G. Fatta, "BodyCloud: Integration of cloud computing and body sensor networks," in *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 851–856, Taipei, Taiwan, 2012.

[9] J. D. Guttman, "Cryptographic protocol composition via the authentication tests," in *The 12th International Conference on Foundations of Software Science and Computational Structures*, vol. 5504, pp. 303–317, York, UK, 2009.

[10] J. D. Guttman and F. J. Thayer, "Authentication tests and the structure of bundles," *Theoretical Computer Science*, vol. 283, no, 2, pp. 333–380, 2002.

[11] K. Han, K. Kim, and T. Shon, "Untraceable mobile node authentication in WSN," *Sensors*, vol. 10, no. 5, pp.4410–4429, 2010.

[12] 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.

[13] 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.

[14] 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.

[15] S. Hyun, P. Ning, A. Liu, and W. Du, "Seluge: Secure and DoS-resistant code dissemination in wireless sensor networks," in *The 7th International Conference on Information Processing in Sensor Networks*, pp. 445–456, St. Louis, MO, 2008.

[16] S. Jiang, J. Miao, and L. Wang, "Mobile node authentication protocol for crossing cluster in heterogeneous wireless sensor network," in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 205–209, Xi'an, China, 2011.

[17] S. Jiang, J. Zhang, J. Miao, and C. Zhou, "A privacy-preserving reauthentication scheme for mobile wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 913782, pp. 1–8, 2013.

[18] D. Kar, R. Tatum, and K. Zejdlik, "MHIP: Effective key management for mobile heterogeneous sensor networks," *International Journal of Network Security*, vol. 15, no. 4, pp. 280–290, 2013.

[19] J. Katz, "Universally composable multi-party computation using tamper-proof hardware," in *26th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 4515, pp.115–128, Springer, 2007.

[20] J. Li, B. Bhattacharjee, M. Yu, and R. Levy, "A scalable key management and clustering scheme for wireless ad-hoc and sensor networks," *Future Generation Computer Systems*, vol.24, no. 8, pp. 860–869, 2008.

[21] 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.

[22] X. Li, X. Lu, J. Ma, Z. Zhu, L. Xu, and Y. Park, "Authentication and key management in 3G-WLAN interworking," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 394–407, 2011.

[23] X. Li, Y. Zhang, X. Liu, J. Cao, and Q. Zhao, "A lightweight roaming authentication protocol for anonymous wireless communication," in *2012 Global Communications Conference (GLOBECOM)*, pp.1029–1034, Anaheim, CA, 2012.

[24] M. E. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet drop in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no.8, pp.3947–3962, 2011.

[25] M. E.Mahmoud and X. Shen, "FESCIM: Fair, efficient, and secure cooperation incentive mechanism for multi-hop cellular networks," *IEEE Transactions on Mobile Computing*, vol. 11, no.5, pp.753–766, 2012.

[26] G. de Meulenaer, F. Gosset, F. X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *The 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB'08)*, pp. 580–585, Avignon, France, 2008.

[27] P. Ning, A. Liu, and W. Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–35, 2008.

[28] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *ACM Wireless Network*, 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," *Sensor*, vol. 10, no. 4, pp. 3718–3731, 2010.

[30] C. Ran, "Universally composable security: A new paradigm for cryptographic protocols," in *42nd Annual Symposium on Foundations of Computer Science*, pp. 136–145, Las Vegas, NV, 2001.
[31] C. Ran, "Universally composable security: A new paradigm for cryptographic protocols," *Cryptology ePrint Archive*, Report 2000/067, 2005.
[32] H. Tan, D. Ostry, J. Zic, and S. Jha, "A confidential and DoS-resistant multi-hop code dissemination protocol for wireless sensor networks," in *The Second ACM Conference on Wireless Network Security*, pp. 245–252, Zurich, Switzerland, 2009.
[33] Z. Wan, K. Ren, and B. Preneel, "A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks," in *The First ACM Conference on Wireless Network Security (ACM WiSec'08)*, pp. 62–67, Alexandria, VA, 2008.
[34] J. Wang, Y. Yu, 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.
[35] L. Wang and Y. Shi, "Patrol detection for replica attacks on wireless sensor networks," *Sensors*, vol.11, no.3, pp. 2496–2504, 2011.
[36] Y. Wang, D. S. Wong, and L. Huang, "One-pass key establishment protocol for wireless roaming with user anonymity," *International Journal of Network Security*, vol. 16, no. 2, pp. 129–142, 2014.
[37] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing sensor networks with public key technology," in *The 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 59–64, Washington, DC, 2004.
[38] G. Yang, "Comments on an anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Transactions on Wireless Communication*, vol. 10, no. 6, pp. 2015–2016, 2011.
[39] G. Yang, Q. Huang, D. Wong, and X. Deng, "Universal authenticated protocols for anonymous wireless communications," *IEEE Transactions on Wireless Communications*, vol.9, no.1, pp.168–174, 2010.
[40] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15.4," *Sensor Network Operations*, Wiley-IEEE Press, pp. 218–237, 2006.
[41] W. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in mobile sensor networks: theory and approaches," *Security and Communication Networks*, vol. 5, no. 5, pp. 496–507, 2012.

Qing-Qing Xie received her B.S. degree from school of computer science and technology, Anhui University in 2012. Now She is working toward PhD degree in Anhui University, China. Her research interests include cryptology and data security.

Shun-Rong Jiang received his M. S. degree in computer science in Jiangsu University, China, in 2012, and now he is studying for his Ph.D degree in Cryptology at Xidian University China,. His research interests include wireless communication security and cryptographic protocols.

Liang-Min Wang received his B. S. degree in computational mathematics in Jilin University, China, in 1999, and the Ph.D degree in Cryptology from Xidian University, China, in 2007. From 2009 to 2010, he was also a visiting scholar in Nanyang Technological University of Singapore. Now he is an associate professor of Jiangsui University. His research interests include security protocols and wireless sensor networks. Currently, he is a senior member of CCF, and a member of IEEE and ACM.

Chin-Chen Chang received his BS degree in applied mathematics in 1977 and the MS degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. During the academic years of 1980-1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the college of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of the R.O.C. From 2002 to 2005, he was a Chair Professor of National Chung Cheng University. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

A Secure and Robust Image Watermarking System Using Normalization and Arnold Scrambling

Dharmalingam Vaishnavi and T. S. Subashini
(Corresponding author: Dharmalingam Vaishnavi)
Department of Computer Science and Engineering, Annamalai University
Annamalai Nagar, Chidambaram, Tamil Nadu, India
(Received Nov. 3, 2014; revised and accepted Jan. 16 & Mar. 4, 2015)

Abstract

This article proposes an image watermarking scheme to improve the robustness and security of watermark against several attacks. To achieve this, image normalization is utilized where affine transformation is applied on the image and which makes it as invariant to geometric transformations. The Lifting Wavelet Transform (LWT) and block based Discrete Cosine Transform (DCT) is applied to the cover image after normalizing an image. Then, the DC coefficients from all blocks are gathered and singular value matrix is constructed using Singular Value Decomposition (SVD). The watermark image is embedded in this singular value matrix after scrambling the image, which increases the security of the proposed scheme. The robustness and invisibleness of the proposed scheme is measured using Peak Signal to Noise Ratio (PSNR), Signal to noise Ratio (SNR) and Structural Similarity (SSIM). The comparison was made with existing schemes and it reveals that the proposed scheme provides high robustness than the existing one.

Keywords: Copyright protection, distortions, image watermarking, scrambling

1 Introduction

The marvelous growth in computer networks and the world wide web coupled with the exponential increase of computer performance, has facilitated the distribution of multimedia data [20]. The extraordinary technical revolution from analog to numerical technology was not achieved without generating anxiety in terms of the protection of the author's rights, since the digital media content, including audio, video and image can be quite easily duplicated, modified and illegally attacked by anyone, and without deterioration of the original image [10]. Thus, it came to be progressively imperative for creators of the computerized media archives to ensure themselves and secure interactive media reports as they were influenced by noteworthy income misfortunes.

Digital watermarks provide a solution to this issue and the exchange of multimedia documents has been since more secure. The digital watermarking was introduced at the beginning of the 1990s, as a second level of technical security protection after encryption. It consists of inscribing invisible secret data into the multimedia document to protect [12]. The key point of digital watermarking is to find the balance between the aspects such as robustness to various attacks and invisibleness. The invisibleness of watermarking is based on the intensity of embedding watermark and the better invisibleness is achieved for less intensity watermark. In general, there is a little compromise between the robustness and invisibleness. Increased robustness require a stronger embedding, which in turn increases the visual degradation of the images [19]. Under the human perception, the digital image watermarking scheme can be classified into two categories: visible and invisible. Visible watermarking in which, the information is obvious in the picture/video. For invisible watermarking, information is added as digital data audio, picture/video, but it cannot be perceived as such. Further, the invisible watermarks are categorized into watermarking techniques as fragile, semi fragile and robust. Fragile/semi-fragile watermark is applied to content authentication and integrity verification because of its sensitiveness to attacks. A robust method is generally used for copyright protection and ownership identification because they are designed to withstand attacks. But the geometric attacks are more complicated to deal with the other kinds of attacks. In this article, a watermarking method is proposed to alleviate the problem of distortion for geometric attacks. The rest of this paper is structured as follows: section 2 gives related works for solution to geometric distortion, section 3 gives the background details of the algorithms used, section 4 illustrates the proposed methodology, section 5 discusses the results and section 6 concludes the paper.

2 Related Works

The work in [4], the original image is transformed into Discrete Wavelet Transform (DWT) domain and a reference sub-image is formed using directive contrast and wavelet coefficient. Then the watermark is embedded by modifying the singular values of reference image using the singular values of the watermark. Many of the transform domain watermarking schemes are proposed by the authors in [3, 16, 23]. In [1] pseudorandom sequence of real number is used as watermark and genetic programming (GP) is used to structure the watermark for enhanced imperceptibility by reflecting the Human Visual System (HVS) aspects. The watermark is detected using correlation. The authors in [6], presented a robust and blind DWT based digital image watermarking scheme. The host image is transformed to wavelet domain and SVD is applied to each sub band. The watermark image is converted to form a new semi binary array and which is inserted into the selected values of SVs of decomposed host image's sub band.

The scheme in [12], the host image is normalized and Harris feature points are extracted to generate some non-overlapped circular regions. The watermark is embedded and extracted into classified regions using the DCT domain. In [9], Zernike transform is applied to the normalized host image to calculate Zernike moments. Dither modulation is adopted to quantize the magnitudes of Zernike moments according to the watermark bit stream. The quality degradation of watermarked image brought about by the embedded watermark is visually transparent.

The work in [8], the first SVs of adjacent blocks of the normalized host image are concatenated to form a singular value (SV) block. DCT is then carried out in these SV blocks. A watermark bit is embedded in the high frequency band of an SVD-DCT block by infringing a specific relationship between two pseudo randomly selected DCT coefficients. The authors in [13], offered a scheme based on logo embedding in DCT domain using image normalization techniques. A visual mask is developed to get maximum watermark embedding with least perceptual degradation. The watermarking structure is based on DCT transform. In [14], visually significant feature points are extracted by end-stopped wavelet. The watermark is embedded in the non-overlapping circular images which are determined by the feature points. These feature points can be used as synchronization marks between watermark embedding and detection. The work in [11], both the normalized host image and watermark image is divided into 8×8 sized block and DCT is applied on each block of host image. Then each watermark block is embedded in the transformed block respectively. The proposed scheme in [18], Blind Normalization Algorithm (BNA) is used to achieve affine invariant wavelet transform. In this, the first step is rotate and scale (RnS) that rotates the signal by a fixed angle θ followed by scale normalization. The second step is the computation of the orientation indicator index (OII). The normalized cover image is wavelet decomposed then the watermark is embedded in DC coefficients of a DCT transformed image. The watermarking schemes in [11, 18, 24, 25] were implemented in DCT and wavelet domains alone and the robustness achieved by these schemes were not that impressive. To improve the robustness further, the proposed scheme combines the LWT and DCT algorithms with Singular Value Decomposition.

3 Background Details

3.1 Discrete Cosine Transform(DCT)

DCT is used to convert the Time domain/spatial domain signal into the frequency domain signal. It is widely used algorithm, due to its compaction and de-correlation properties. The DCT of a given matrix gives the frequency coefficients in the form of another matrix and it is scattered into two: DC and AC coefficients. The left top corner element (zero frequency) is called as DC coefficient which is perceptually significant and which aids to enhance the robustness of the watermark [26].

3.2 Lifting Wavelet Transform (LWT)

LWT with standard 4-tap ortho normal filter with two vanishing moments is used for digital image watermarking. This algorithm consists of following three steps and is given by [21].

1) Split: It splits an input signal $x(n)$ into even and odd samples:

$$x_e(n) = x(2n), x_o(n) = x(2n+1) \tag{1}$$

2) Prediction: It denotes high frequency components of $x(n)$. It takes a difference, between the prediction value of even sample and the original value of odd sample and is denoted as detail signal $d(n)$.

$$d(n) = x_o(n) - P[x_e(n)] \tag{2}$$

where P is prediction operator.

3) Update: Update: It updates the even samples using $d(n)$ and it denotes low frequency components of $x(n)$ and is denoted as $c(n)$.

$$c(n) = x_e(n) + U[d(n)] \tag{3}$$

where U is update operator.

In comparison with general wavelets, reconstruction of image by lifting wavelet is flawless because, it increases smoothness and reduces aliasing effects. It reduces loss of information, increases intactness of the embedded watermark in the image and helps to increase the robustness of the watermark [23].

3.3 Singular Value Decomposition (SVD)

SVD is an optimal matrix decomposition technique and it packs the maximum signal energy into as few coefficients as possible. It has the ability to adapt to the variations in local statistics of an image [5]. The main features of SVD under the perspective of image processing are as follows:

- The quality of the reconstructed image will not degrade a considerable measure, even if ignoring the small SV's in the reconstruction of images.
- The SVs have very good stability, i.e. When a small annoyance is added to an image, the SVs do not vary rapidly.

An image of a real matrix with the size of m×n can be decomposed as: $A = U * S * V'$. Where U is a m×m unitary matrix, S is a m×n matrix with nonnegative numbers on the diagonal and zeros on the off diagonal, and V' denotes the conjugate transpose of V is a n×n unitary matrix. The unitary matrix U and V represent the geometry of an image. The nonnegative components of S represents the luminance value of the image.

3.4 Image Normalization

Image normalization is used to perform watermark embedding and extraction in its original coordinate system by affine transforming the image. The transform parameters are estimated from the geometric moment of the image. Therefore, the image which is invariant to any affine distortions of the image [2, 15]. It will ensure the integrity of watermark, even if, the watermarked image undergoes affine geometric attacks and which increases robustness of watermark [7]. The normalization procedure is composed of the following steps [13].

Step 1: To obtain the translation invariance, shifting the cover image to its central point, image center for $f(x, y)$ is determined by the equation $\begin{pmatrix} x_a \\ y_a \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} - d$, where, $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $d = \frac{d_1}{d_2}$ with: $d_1 = \frac{m_{10}}{m_{00}}, d_2 = \frac{m_{01}}{m_{00}}$. Where m_{00}, m_{01}, m_{10} are geometric moments of an image and let $f_1(x, y)$denotes the resulting center image.

Step 2: Apply shearing transform on $f_1(x, y)$ in the X-direction with matrix denoted $A_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ by $f_2(x, y) = A_x[f_1(x, y)]$. The value of β can be calculated using following equation $\mu_{30} = \mu_{30} + 3\beta\mu_{21} + 3\beta\mu_{12} + \beta^3$.

Step 3: Apply transform to $f_2(x, y)$ in the Y-direction with the matrix is denoted $A_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$ by $f_3(x, y) = A_y[f_2(x, y)]$. The value of γ can be calculated using following equation $\mu_{11} = \gamma\mu_{20} + \mu_{11}$.

Step4: Apply Scale $f_3(x, y)$ in both X and Y directions such that $A_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$ andthe resulting image is denoted by $f_4(x, y) = A_s[f_3(x, y)]$. The value of parameters α and δ are determined by its moments $\mu_{50} > 0, \mu_{05} > 0$ respectively.

Where, μ is central moment & m is the geometric moment of an image with $p + q$ order and $p, q = 0, 1, 2, ...$

3.5 Arnold Scrambling

Scrambling is a pretreatment stage of watermarking and which makes the meaning full image as meaningless one. It is an essential issue to have the spatial correspondence decreased between the host image and the embedded watermark [22]. The 2-dimensional Arnold scrambling transformation is defined as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} modN \tag{4}$$

$$x, y \in 0, 1, 2, ...N - 1$$

Wherein, x and y is the pixel coordinates of the original space: x' and y' is the pixel coordinates, after iterative computation scrambling; N is the size of the image, also referred to as a step number. By the above formula the corresponding inverse transform formula can be obtained:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} modN \tag{5}$$

$$x', y' \in 0, 1, 2, ...N - 1$$

It is easy to restore the original initial state according to the corresponding iterations [17]. Arnold transformation is cyclical, when iterate to a step, it will regain original image. It is not able to restore the image without knowing about the cycle and iterations. Thus it makes security of the embedded watermark strongest.

4 Proposed Method

The watermarking scheme consists of watermark embedding and extraction processes. These two processes of proposed scheme are illustrated in the following subsections.

4.1 Watermark Embedding

The basic idea of invisible image watermark is to embed the watermark on the cover image without damaging its visual content. The proposed watermark embedding process is shown in Figure 1 and the steps involved in it are as follows:

Step 1: The normalization procedure is applied to the cover image.

Step 2: Normalized cover image is converted into four sub bands (LA, LH, LV& LD) of wavelet coefficients by applying single level LWT with Haar wavelet scheme.

Step 3: The wavelet coefficients of sub band is selected and divided as 8×8 non-overlapping sub blocks.

Step 4: Then the forward DCT is applied to each non overlapping block. Then, DC coefficient from each DCT transformed block is retrieved to form a matrix of DC's and Singular values are extracted using SVD.

Step 5: The watermark image is embedded on the matrix of singular values using controlling parameter called gain factor (g), after applying the Arnold scrambling.

Step 6: The modified SVs are replaced with original DC coefficients of each DCT transformed block and then inverse DCT is applied to each sub blocks.

Step 7: The all 8×8 non-overlapping sub blocks are organized as a single block matrix and the image is reconstructed using Inverse LWT with one modified and other three unmodified sub bands.

Step 8: The inverse normalization procedure is applied to get the watermarked image.

4.2 Watermark Extraction

The steps involved in the watermark extraction process are as follows:

Step 1: First, the normalization procedure is applied to the watermarked image then it is decomposed using single level LWT to get the sub bands of coefficients.

Step 2: The sub band, which has chosen for embedding process is selected and divided as 8×8 non-overlapping sub blocks to transform it to DCT coefficients.

Step 3: The DC coefficients from each DCT transformed block is retrieved and SVD is applied to extract the scrambled watermark coefficients.

Step 4: Then the inverse scrambling procedure is applied to obtain a meaning full recovered watermark image.

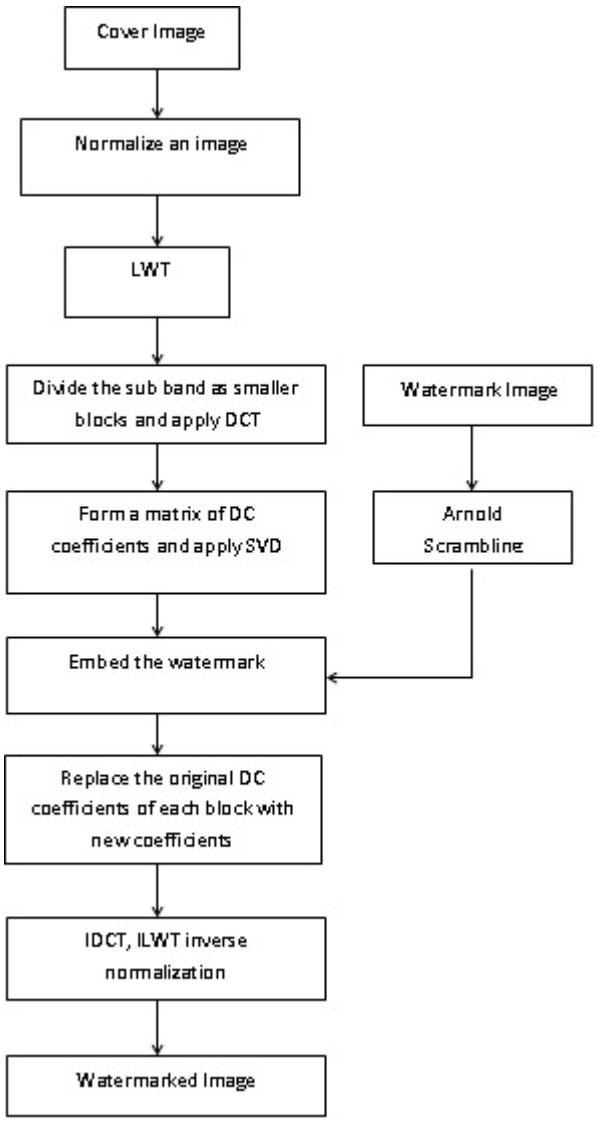


Figure 1: The Proposed scheme

Figure 2: Sample Cover and Watermark images

5 Results and Discussions

The experiment was done by Mathworks MATLAB version 12b. The single level LWT with Haar wavelet is applied to an input image. The watermark is embedded into the wavelet of each sub band. The input of cover and watermark images are taken as a grayscale image whose resolution is 256×256 and 32×32 respectively. Figure 2 shows the sample cover image flower and the watermark K logo. Figure 3 shows watermarked flower image and the recovered K logo watermark.

In order to test the fidelity and robustness of proposed method, the metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) and Signal-to-Noise Ratio (SNR) are calculated by Equations (6)-(9). Here, O & W represents the Original and Watermarked

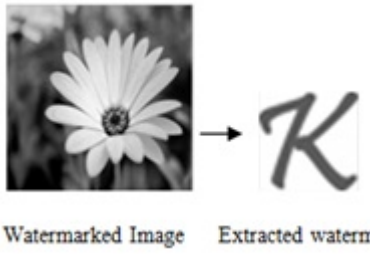


Figure 3: Watermarked image and Extracted watermark

and or recovered images.

$$PSNR = 10log_{10}\left(255^2/MSE\right) \quad (6)$$

$$MSE = \frac{\sum[O(m,n) - W(m,n)]}{M \times N} \quad (7)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (8)$$

where,

- μ_x, μ_y is the average of x,y respectively;
- σ_x^2, σ_y^2 is the variance of x and y;
- $\sigma_{x,y}$ is covariance of x and y;
- $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ are two variables to stabilize the division with weak denominator;
- L is the dynamic range of the pixel values $k_1 = 0.001$ *and* $k_2 = 0.03$ by default.

$$SNR = \frac{\sum_{i=o}^{M-1}\sum_{j=0}^{N-1}[O(i,j)^2]}{\sum_{i=o}^{M-1}\sum_{j=0}^{N-1}[(O(i,j)^2 - W(i,j))^2]} \quad (9)$$

To experimentally ascertain the robustness and fidelity, the proposed approach is assessed with the watermarked image against the following attacks: Rotation, Scaling, Translation, X-Shearing, Y-shearing, noise, Contrast Increment (Cont Inc) and Histogram Equalization (Hist Eq) and the results are exposed in the Figure 4 & Figure 5. It depicts that the proposed system is resilient to all the attacks.

The fidelity and robustness of proposed scheme subject to different sub bands for the flower image is embedded with k logo watermark, is measured by PSNR calculation and is given in Table 1. It shows that the PSNR value of robustness is greater in LD sub band when compared with the other sub bands. Further, Table 2 displays the robustness of the proposed system using PSNR, SSIM and SNR measures for LD sub band subjecting to the geometric and other kinds of attacks for the gain factor of 0.5. It shows that the robustness (PSNR value) acheived is excellent for the geometric attacks scaling (46.6051), rotation (46.6092), and translation (46.6204), X- shearing (59.9161) and Y-shearing (60.1973).

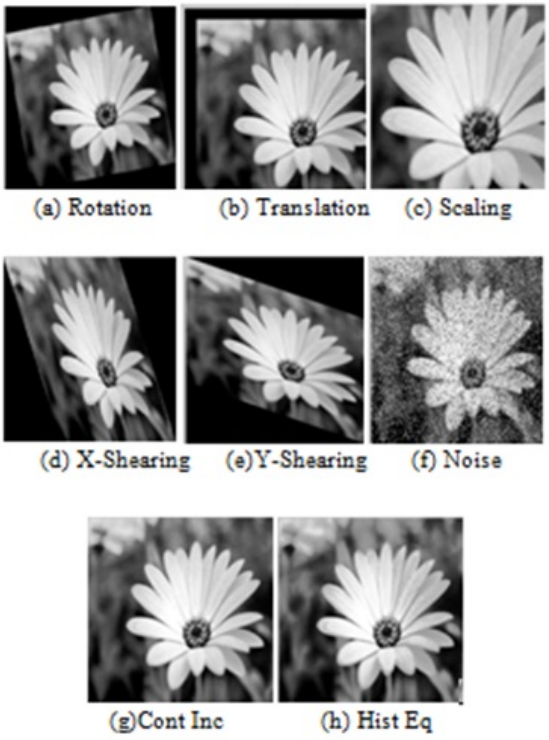


Figure 4: Flower image embedded with K logo watermark subjected to various attacks a) rotation, b) translation, c) scaling, d) X-Shearing, e) Y-shearing, f) noise, g) Cont. Inc and h) Hist Eq

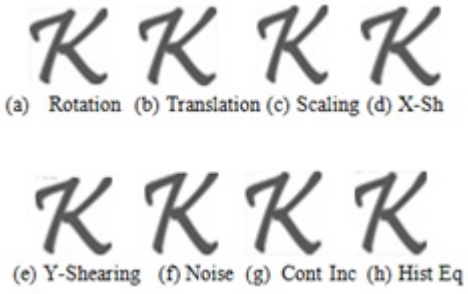


Figure 5: K logo watermark extracted from flower image after subjecting it to various attacks a) rotation, b) translation, c) Scaling, d) X-Shearing, e) Y-shearing, f) noise, g) Cont Inc and h) Hist Eq.

Figure 6: Cover image man and watermark CS logo image

5.1 Comparison of Proposed Scheme with BNA Scheme

In order to justify the results, the proposed scheme is compared with the existing system Blind Normalization

Table 1: Fidelity and robustness (PSNR value) of the flower image watermarked with K logo watermark in different sub bands

Gain Factor	LA	LH	LV	LD
Fidility				
0.1	38.9936	39.5433	39.6337	39.8160
0.5	31.2529	34.6081	35.1472	37.2655
1	26.2422	30.1644	31.1007	33.6606
Robustness				
0.1	20.9747	37.9505	35.1736	49.7414
0.5	17.9527	39.8228	46.3689	65.6337
1	24.9196	37.3635	46.3348	71.2505

Table 2: Robustness of extracting K logo watermark for gain factor=0. 5

Attacks	PSNR	SNR	SSIM
Rotation	46.6092	59.3065	0.9996
Translation	46.6204	62.4176	0.9998
Scaling	46.6051	58.0547	0.9996
X-Shearing	59.9161	58.3359	0.9996
Y-Shearing	60.1973	59.3065	0.9996
Noise	34.7532	32.8918	0.9996
Cont. Hist.	66.2200	64.3586	0.9996
Hist Eq.	62.1603	60.2989	0.9996

Figure 7: Cover image man embedded with CS logo watermark with various attacks a) Noise, b) Rotation, c) Cont Inc, d) Hist Eq, e)Scaling

Algorithm (BNA) based DWT- DCT watermarking[21]. The sample images in the BNA method namely man and CS logo is used as the benchmark image to do the comparative study about the performance of the proposed method and is shown in Figure 6. The Figure 7 displays the watermarked man image after the geometric attacks (Rotation & scaling) and other attacks (Noise, Cont Inc & Hist Eq).

Figure 8 displays the extracted watermark CS logo subjecting to the attacks. For comparison purpose, the proposed system is tested with the same gain factors namely

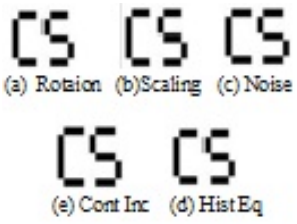


Figure 8: CS logo watermark extracted from Man image after subjecting it to various attacks a) Noise, b) Translation, c) cont Inc, d) Histogram Equalization, e) Scaling

0.1, 0.5 and 1. Table 3 & Table 4 showing the comparison of fidelity and robustness of proposed method with BNA method for different sub bands with gain factors. It demonstrates that the robustness as well as the fidelity of the proposed method is far better compared to the BNA method. The improved PSNR and SNR values of the proposed method show that the proposed method is highly robust. It also shows that the LD sub band gives better PSNR and SNR values out of four sub bands. The results also depict that the watermark can be embedded in the LD sub band without any deterioration in the image quality (fidelity) and as well as it is robust to any kind geometric attacks. Further, the Table 3 also reveals that the fidelity is better for smaller gain factor.

The Figures 9-12 graphically shows that the comparison of robustness of the proposed system with BNA method for the different wavelet sub bands. Table 5 also gives a better robustness for LD sub band for the attacks with 0.5 gain factor and which is graphically shown in the Figure 13. It also shows that proposed method is highly improved compared to BNA method.

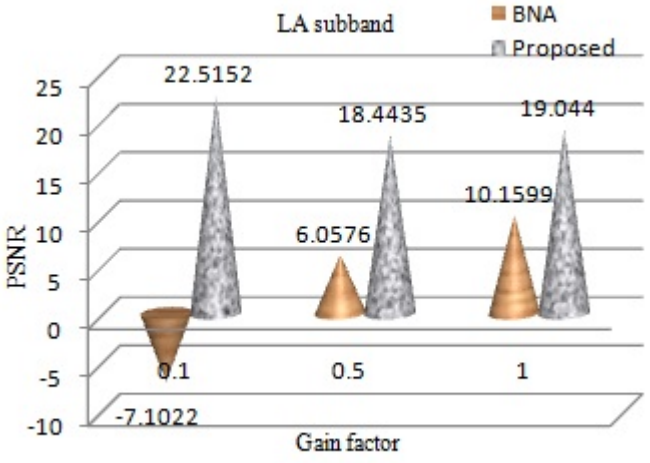


Figure 9: Gain factor vs PSNR (Robustness) for LA sub band of cover image man embedded with CS logo watermark

5.2 Robustness Against Multiple Attacks

The robustness of the proposed system is also tested to ascertain how it handles multiple geometric attacks at a

Table 3: Comparison of Fidelity for watermarked man image for different wavelet sub bands

Sub Band	Gain Factor	**Fidility**			
		BNA Method		**Proposed Method**	
		PSNR	**SNR**	**PSNR**	**SNR**
LA	**0.1**	13.7642	1.3316	34.1176	21.7089
	0.5	13.7528	1.3316	29.9730	17.5492
	1	13.6901	1.2948	26.0615	13.0445
LH	**0.1**	33.9066	-0.0668	34.2859	21.8650
	0.5	32.1404	-0.0852	18.5312	18.5312
	1	29.6951	-0.1007	29.7522	15.9511
LV	**0.1**	33.8943	-0.0672	32.7276	20.3010
	0.5	32.0385	-0.0847	31.8376	19.4194
	1	29.6203	-0.0995	29.9251	17.5330
LD	**0.1**	33.9892	-0.0668	34.7744	19.3031
	0.5	32.4560	-0.0864	32.7048	17.2510
	1	30.2078	-0.1022	29.2401	15.0430

Table 4: Comparison of Robustness of the proposed method with BNA method for different wavelet sub bands

Sub Band	Gain Factor	**Fidility**			
		BNA Method		**Proposed Method**	
		PSNR	**SNR**	**PSNR**	**SNR**
LA	**0.1**	-7.1022	-43.9509	22.5152	4.4709
	0.5	6.0576	-26.9779	18.4435	9.2920
	1	10.1599	-24.0186	19.0440	7.7655
LH	**0.1**	13.5618	-4.9816	31.5569	19.9601
	0.5	13.5498	4.1096	43.5801	34.3042
	1	13.4536	7.3074	42.0741	32.8275
LV	**0.1**	13.6057	-3.9755	37.6449	27.3884
	0.5	13.5384	3.8456	51.1598	41.9306
	1	13.4780	7.3965	58.7038	49.4626
LD	**0.1**	13.4157	1.1388	46.2885	37.0577
	0.5	13.4139	12.1995	57.9158	48.6794
	1	13.3901	14.4298	66.2543	57.0100

Table 5: Comparison of robustness (LD sub band) of proposed method with BNA method for different attacks (g=0.5)

Methods	**BNA Method**		**Proposed Method**	
Attacks	**PSNR**	**SNR**	**PSNR**	**SNR**
Noise	13.3336	4.9786	35.4550	26.2351
Rotation	13.4263	10.9554	39.3071	49.7238
Cont. Inc.	13.4329	10.1274	55.5388	46.3165
Hist. Eq.	13.4643	8.3710	55.1915	45.9938
Scaling	13.4925	7.5581	39.2162	44.8618

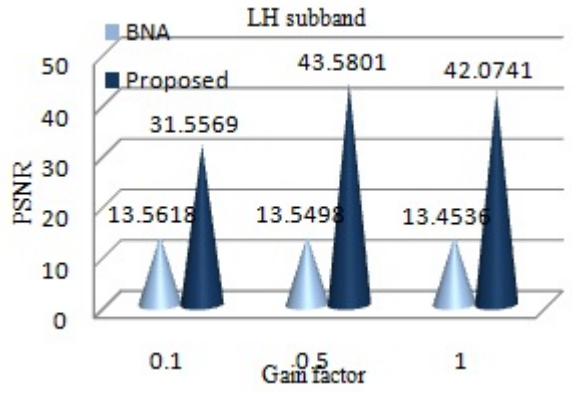


Figure 10: Gain factor vs PSNR (Robustness) for LH sub band of cover image man embedded with CS logo watermark

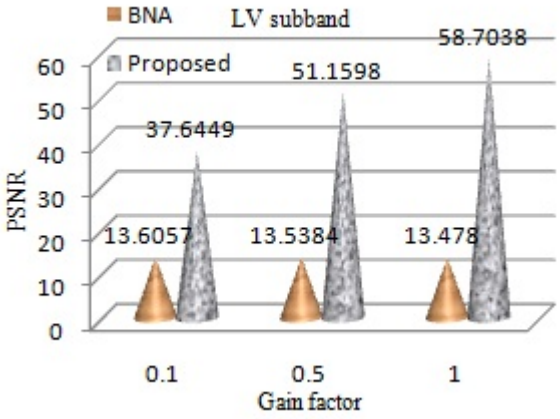


Figure 11: Gain factor vs PSNR (Robustness) for LV sub band of cover image man embedded with CS logo watermark

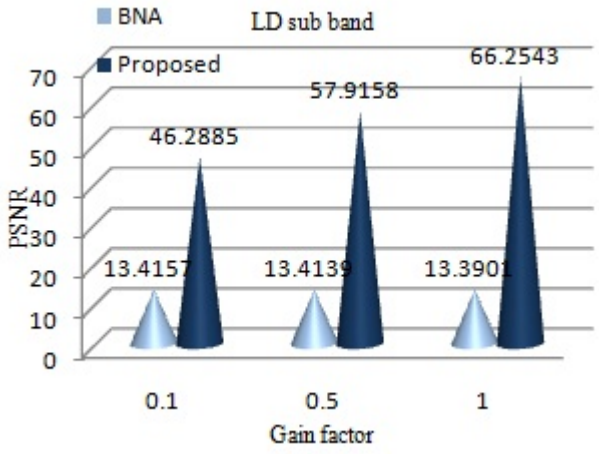


Figure 12: Gain factor vs PSNR (Robustness) for LD sub band of cover image man embedded with CS logo watermark

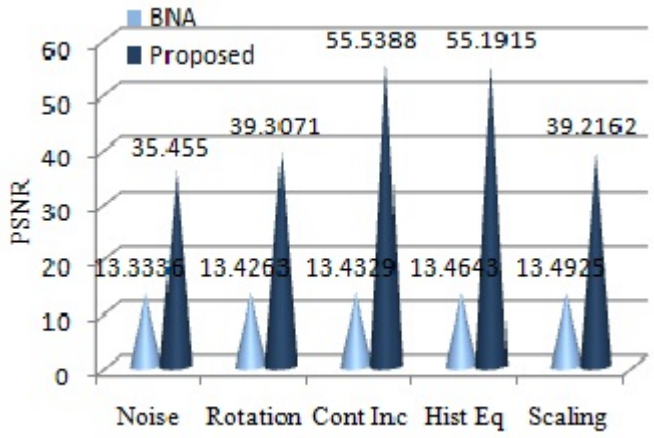


Figure 13: Robustness under the attacks for LD sub band for cover image man embedded with CS logo watermark

time. Hence, the various combinations of geometric attacks were applied to watermarked images and the watermark was recovered after the attacks. The images are taken from the standard image processing database which was embedded with the watermark shown in Figure 2. The watermark was recovered after the attacks and results are shown in Figure 14. The experiment is repeated with 20 images from the database and the mean quality measures such as PSNR and SNR tabulated in Table 6. Table 6 reveals that the proposed method is resilient to combination of various attacks also.

Table 6: Robustness for combination of geometric attacks for gain factor 0.5

Combination of Attacks	**Quality Metrics**	
	PSNR	**SNR**
Rotation +Translation	50.2126	38.5376
Translation + Scaling	41.6941	29.1566
Rotation+Scaling	46.3957	34.6203
Rotation+Translation+Scaling	46.1827	34.1939
X&Y directional Shearing	49.3310	37.8248
Rotation+Translation +X & Y Directional Shearing	50.4827	39.0628
Translation+Scaling +X & Y Directional Shearing	40.8880	28.1711
Rotation+Scaling +X & Y Directional Shearing	45.3821	33.6296
Rotation+Translation+Scaling +X & Y Directional Shearing	41.3514	28.6097

6 Conclusion

In this paper, a secure and robust watermarking scheme is developed for geometric distortion and other several attacks using image normalization. The robustness of wa-

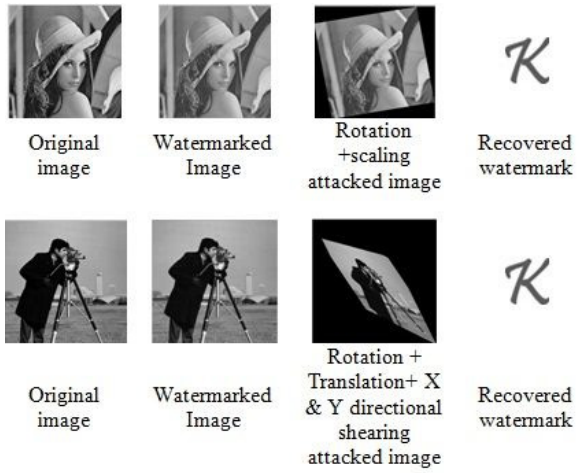


Figure 14: Results of Combination of Gometric Attacks

termark is improved by combining the algorithms namely LWT & DCT with Singular Value Decomposition. The security of proposed scheme is also achieved by scrambling the watermark image. Then, the proposed method was tested in different wavelet sub bands and various gain factors. The results from the test illustrates that the watermark embedded in the LD sub band is highly robust to geometric and various kinds of attacks. The performance of the proposed system is also compared with the results of the BNA method for various gain factors and it also reveals that the proposed system is superior and LD sub band is highly to the attacks.

References

[1] A. Abbasi, W. C. Seng, and I. S. Ahmad, "Multi block based image watermarking in wavelet domain using genetic programming," *International Arab Journal of Information Technology*, vol. 11, no. 6, pp. 582–589, 2014.

[2] Y. S. Abu-Mostafa and D. Psaltis, "Image normalization by complex moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 1, pp. 46–55, 1985.

[3] M. S. Arya, R. Siddavatam, and S. P. Ghrera, "A hybrid semi-blind digital image watermarking technique using lifting wavelet transformsingular value decomposition," in *IEEE International Conference on Electro/Information Technology (EIT'11)*, pp. 1–6, 2011.

[4] G. Bhatnagar and B. Raman, "A new robust reference watermarking scheme based on DWT-SVD," *Computer Standard & Interfaces*, vol. 31, pp. 1002–1013, Sept. 2009.

[5] D. S. Chandra, "Digital image watermarking using singular value decomposition," in *The 45th IEEE Midwest Symposium on Circuits and Systems (MWSCAS'02)*, vol. 3, pp. III–264, 2002.

[6] H. Danyali, M. Makhloghi, and F. A. Tab, "Robust blind DWT based digital image watermarking using singular value decomposition," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 7, pp. 4691–4703, 2012.

[7] P. Dong, J. G. Brankov, N. P. Galatsanos, Y. Yang, and F. Davoine, "Digital watermarking robust to geometric distortions," *IEEE Transactions on Image Processing*, vol. 14, pp. 2140–2150, 2005.

[8] S. W. Foo and Qi Dong, "A normalization-based robust image watermarking scheme using SVD and DCT," *International Scholarly and Scientific Research & Innovation*, vol. 4, no. 1, pp. 753–758, 2010.

[9] S. W. Foo, Qi Dong, "A normalization-based robust watermarking scheme using zernike moments," *World Academy of Science, Engineering and Technology*, vol. 35, pp. 508–513, 2009.

[10] Li C. Huang, L. Yu Tseng, and M. S. Hwang, "The study of data hiding in medical images.," *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[11] A. Kumar, A. Kr Luhach, and D. Pal, "Robust digital image watermarking technique using image normalization and discrete cosine transformation," *International Journal of Computer Applications*, vol. 65, no. 18, pp. 5–13, 2013.

[12] K. Loukhaoukha, J. Y. Chouinard, and M. H. Taieb, "Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 4, pp. 303–319, 2011.

[13] F. K. Mohamed and R. Abbes, "RST robust watermarking schema based on image normalization and DCT decomposition," *Malaysian Journal of Computer Science*, vol. 20, no. 1, pp. 77, 2007.

[14] I. Nasir, F. Khelifi, J. Jiang, and S. IPSON, "Robust image watermarking via geometrically invariant feature points and image normalisation," *IET Image Processing*, vol. 6, no. 4, pp. 354–363, 2012.

[15] S. C. Pei and C. N. Lin, "Image normalization for pattern recognition," *Image and Vision Computing*, vol. 13, no. 10, pp. 711–723, 1995.

[16] C. Sharma and D. Prashar, "Visible and invisible watermarking methods for quality loss of data," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, no. 3, pp. 57–63, 2012.

[17] C. Song, S. Sudirman, and M. Merabti, "Recent advances and classification of watermarking techniques in digital images," in *Proceedings of Post Graduate Network Symposium*, pp. 1–6, 2009.

[18] T. Sridevi and V. V. Kumar, "A robust watermarking algorithm based on image normalization and dc coefficients," *International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 226–232, 2011.

[19] S. S. Sujatha and M. M. Sathik, "A novel DWT based blind watermarking for image authentication," *International Journal of Network Security*, vol. 14, no. 4, pp. 223–228, 2012.

[20] B. Surekha and G. N. Swamy, "Sensitive digital image watermarking for copyright protection," *International Journal of Network Security*, vol. 15, no. 2, pp. 113–121, 2013.

[21] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, 1996.

[22] V. Prasad, R. Kurupati, "Secure image watermarking in frequency domain using arnold scrambling and filtering," *Advances in Computational Sciences and Technology*, vol. 3, no. 2, pp. 236–244, 2010.

[23] D. Vaishnavi and T. S. Subashini, "A novel approach to improve invisibleness and robustness of a digital watermark in copyrightprotection," *International Journal of Computer Applications*, vol. 71, pp. 7–13, June 2013.

[24] D. Vaishnavi and T. S. Subashini, "A robust image watermarking for geometric distortion using dc coefficients," *International Journal of Applied Engineering Research*, vol. 9, no. 21, pp. 4895–4800, 2014.

[25] D. Vaishnavi and T. S. Subashini, "An image watermarking scheme resilient to geometric distortions," in *Power Electronics and Renewable Energy Systems*, LNCS 326, pp. 1225–1233, Springer, 2015.

[26] B. Wang, J. Ding, Q. Wen, X. Liao, and C. Liu, "An image watermarking algorithm based on DWT, DCT and SVD," in *IEEE International Conference on Network Infrastructure and Digital Content (ICNIDC'09)*, pp. 1034–1038, 2009.

Dharmalingam Vaishnavi received her BE (IT) degree in the year 2009 and ME (CSE) degree in the year 2011 from Annamalai University. She was worked as Assistant Professor at Anjalai Ammal Mahalingam Engineering College, Thiruvarur District, Tamil nadu, India. Currently, she is pursuing Doctor of philosophy in Annamalai University. Her area of research is Image watermarking and digital forensics. She has published 9 International Journals and Conferences.

T. S. Subashini received her B.E (CSE) degree in the year 1991 from Bharath Engineering College, Chennai. In the same year she was appointed as Hardware Service Engineer in Sterling Computers Chennai and in 1996 she joined Annamalai University. She was sponsored by the University under Quality Improvement Programme (QIP), to pursue ME (CSE) at Anna University, Chennai 2001. She gained her doctoral degree in Computer Science and Engineering from Annamalai University in 2011. Her area of doctoral research is Medical Image Analysis. She has published over 35 research papers in international journals and conferences. She is now working on UGC sanctioned research project worth Rs. 11.4 lakhs on Breast Cancer. Her research interests include Image and Video processing, Computer Vision and Pattern Classification.

A Lightweight Threat Detection System for Industrial Wireless Sensor Networks

Yosra Ben Saied[1], Alexis Olivereau[2]
(Corresponding author: Yosra Ben Saied)
RAMSIS Group, CRISTAL Lab, National School of Computer Science, Tunisia[1]
CEA, LIST, Communicating Systems Laboratory, Gif-sur-Yvette, France[2]
(Email: yosra_bensaied@yahoo.fr)
(Received Aug. 13, 2015; revised and accepted Sept. 29, 2015)

Abstract

Wireless Sensor Networks (WSNs) led the way to new forms of communications, which extend today the Internet paradigm to unforeseen boundaries such as eHealth, intelligent building or smart grid, to name a few. The legacy industry, however, is slower to adopt this technology, mainly for security reasons. Self-managed security systems allowing a quicker detection and better resilience to attacks, may counterbalance this reluctance. In this article, we propose a hybrid threat detection system that involves both centralized decision and local, per-cluster, work repartition and that is designed to run on top of industrial wireless sensor networks. Compared with the literature, we prove that this system is suitable for architectures mainly composed of constrained and sleeping devices, for which it achieves a fair level of autonomous security without prohibitively drawing out energy resources.
Keywords: Energy efficiency, hybrid approach, intrusion detection, resource constraints, wireless sensor network

1 Introduction

Wireless Sensor Networks (WSNs) changed the Internet communication paradigm by introducing unattended devices in a hitherto human-centric model. WSNs filled the gap between the physical world and the Internet by offering the ability to monitor in real time a wide range of physical values through connected devices.

Initially built around very simple entities, technologies and architectures, WSNs benefitted from a strong interest from the international research community, which designed in a few years a wide variety of new components and systems. Dedicated radio technologies and routing protocols were specified, as well as novel, complex architectures. These architectures evolved from a basic, unidirectional topology made of multiple sensor nodes pushing data to a server through a sink node, to more advanced systems featuring bi-directionality. These new systems paved the way towards the emerging Machine to Machine (M2M) and Internet of Things (IoT) frameworks.

This gain of maturity of WSN technologies accelerates their adoption in the industry, and this adoption is all the quicker as WSNs answer to classical needs of industrial scenarios: monitoring of physical values, asset supervision and facilities surveillance are all key requirements in these scenarios, for which dedicated sensor nodes are available. Introducing actuators along with sensor nodes allows for more complex operations such as asset control and better production chain automation. Finally, the addition of more complex entities, such as indoor positioning nodes and mobile devices, to an existing sensor topology can give rise to new services, also profitable for the industry. For example, better worker safety can be achieved if the production chain is able to detect a worker's presence in an unauthorized area, or to determine that a certain user is not wearing adequate protection suit.

However, even though cost effective devices and energy-efficient technologies and protocols are available, the underlying security question impedes the use of WSNs in the most critical industrial scenarios. The inherent vulnerability of WSN nodes, which is due to their exposed location and their use of wireless communications, is such that a WSN has to mimic all security features from the legacy Internet, while also adding specific use cases and taking into account the strong shortcomings of the WSN nodes. Yet, strong security is often contradictory with limited resources. Furthermore, the unattended nature of WSNs and their autonomous decision taking upon a context change put them at risk of triggering harmful behavior, if misled by an attacker.

We propose to leverage on this WSN autonomy to provide it with the ability to detect new threats and react to them in the most appropriate manner. Contrary to existing work, the proposed threat detection system is lightweight enough to be run on resource-constrained sensor nodes. It greatly improves the resilience of the industrial WSN, without bringing in excessive energy con-

sumption. Our solution is based on a partly centralized architecture and specifies new roles for WSN entities, in accordance with their status and capabilities.

The remainder of this paper is organized as follows. Section 2 quickly describes challenges related to the development of a threat detection framework for WSNs and highlights the limitations of existing approaches. Section 3 describes the proposed threat detection solution and its implementation on physical sensor devices. We assess its performance when compared to main state of the art solutions in Section 4. Section 5 concludes the paper.

2 Problem Statement and Related Work

The need for a threat detection system arises from the need to protect WSNs against attacks. Obviously, we assume that WSNs feature dedicated authentication and access control routines, key establishment functions. The security here is provided by the use of cryptographic primitives, which protects the WSN from external attacks. However, these security functions fail to protect the network against an authenticated attacker from inside the network: a sensor node owning legitimate cryptographic keys can easily launch an internal attack inside the group. It may be reluctant to make its resources available to other nodes as part of a cooperative act. Therefore, such a node may prefer to behave at times in a selfish manner and refuse to cooperate in packet forwarding in order to maximize its energy savings. On the other hand, a legitimate sensor node can also act maliciously by dropping packets, delaying the transmission or sending packets through a different route than planned. Such an internal attacker can only be detected through behavioral analysis mechanisms. The latter track the system behavior and interactions between nodes to detect threat attempts and/or occurrences. Once a security anomaly is detected, a reaction mechanism is launched to take security and service repair measures.

While highly relevant to the autonomous topologies that wireless sensor networking involves, threat detection raises issues with respect to its adaptability to these domains. It challenges indeed the constrained nodes' limited energy resources by involving complex processing and high resource consumption.

Most of existing researches on threat detection systems propose to mimic models proposed for Mobile Ad hoc NETworks (MANETs) in order to counter internal attacks inside WSNs [4, 8, 9]. These threat detection models rely on monitoring agents deployed on each sensor node. The goal of these local agents is to track the traffic within their radio range and detect misbehaving nodes causing routing disruptions. Authors in [10] define a threat detection system based on monitoring agents. They distinguish between local agents (able to process only the packet they actively forward) and global agents. Global agents act as watchdogs by monitoring nearby traffic, especially when they determine how a forwarding node has behaved (e.g. difference, delay and loss between an incoming packet at a neighbor). In addition to regularly monitoring global agents, the authors introduce the concept of spontaneous watchdog behavior: a node determines from a sensed packet that it can be in position to act as a watchdog for this packet. This node will first identify the number n of its neighbors that could also act as watchdogs for the same packet, and turns on its watchdog behavior for that packet with a one-in-n probability. However, local passive monitoring has been shown to draw as much power from the sensor node transceiver as data reception [13]. Ioannis et al. in [7] also apply the watchdog behavior, with a higher emphasis on countering malicious monitoring nodes. To that aim, cooperation through majority is achieved, relying on an encrypted flag. That means that a node will be classified as malicious only if a majority of its neighbors flag it as such. The problem with the use of watchdog behavior is that watchdog nodes will drain their resources quickly.

Butun et al. in [3] review existing intrusion detection systems in WSNs and highlight the fact that IDSs designed for MANETs cannot be applied to WSNs directly. Authors consider that resource constraints characterizing the sensor nodes should be taken into account for the design of an IDS adapted to WSNs. Also, the fact that most of sensor nodes are most of the time in sleeping mode makes the operation of this security system more complex, because synchronous node actions might prove difficult, if not impossible, to achieve.

Other researches recognize that designed IDSs need to spend the least amount of energy as possible to spare enough energy for the crucial operations of the WSN. For this reason, Huang and Lee propose in [6] an energy-efficient monitoring system that fit the energy constraints of sensor nodes. They select a single node to perform monitoring at a given time within a given cluster. This node is designated through a fair election process. Authors claim that they obtain much higher energy efficiency at an equivalent security level compared with systems where all nodes perform monitoring at the same time. They also provide a detailed set of rules that allow detecting classical attacks expected to occur within a wireless sensor network. Authors in [1, 12, 14] propose hierarchical trust management systems for WSNs to detect selfish and malicious nodes. In these approaches, regular sensor nodes do not participate in the global decision making process. Only Cluster Heads (CHs) are responsible for the global decision making process and the response. The main reason for this is to reduce the energy consumption. They wanted to conserve the energy of the majority of sensor nodes, by simply assigning them as subordinates under CHs.

These cluster based detection schemes consider that a node is periodically elected to be the monitoring agent within each cluster. However, the election process could be heavy for constrained nodes -as demonstrated in what

follows. In the same context, Butun et al. in [3] underline that clustering based IDSs may consume considerable amount of the WSN's energy through exchanged messages between nodes to periodically elect new monitoring agents.

3 Solution Description

3.1 Overview

Considering the inadequacies of existing approaches described in the previous section, we propose an efficient threat detection system for WSNs that aims at meeting two key requirements. Considering the energy constraints of sensor nodes will be the first key requirement for the design of our system. Processing within sensor nodes will be minimized in order to increase their lifetime. So that, most of required operations are delegated to the server side and the charge on the sensor node will be kept light, which leads to extend the network lifetime. Offloading the charge from the constrained nodes to the server is not the only advantage of this approach. Having to send its observations to the central server to take decisions, a malicious node would not be in position to propagate false assessments about specific victims. With a central entity responsible for decision making, it becomes a common profit for all nodes to provide reliable evidences since false ones can globally affect decision making at the central entity, and could eventually be detrimental to the attacker itself.

The second important key requirement is that a sensor node is often in a sleeping mode, so that considering that it is able to perform synchronous actions with other nodes for threat detection as proposed in previous works would be difficult to achieve. This problem will be carefully taken into account for the design of our solution. Sensor nodes will wake up only to collect data and perform autonomous security-related actions, then revert back to sleep mode.

3.2 Network Model

The wireless sensor network is supposed to be divided into zones. Each zone contains one or more clusters and each cluster contains one or more sensor nodes. Zones and clusters have different criticalities, different security levels and different security policies. It is also assumed that sensor nodes within the same cluster can communicate with each other. In addition, we assume that the awake time is negligible compared to the sleep time. Finally, we exclude any form of synchronization between nodes.

In a WSN, there are two types of nodes: sensing devices and gateways (or servers). The sensing devices are simple nodes equipped with radio interfaces only. The gateways are equipped with an Ethernet port to communicate with remote application servers and deliver collected data. The Gateway has more computing capacities and unlimited energy resources, as compared with a constrained sensor node. Since these nodes are Ethernet-connected, assuming that they are also connected to a power supply seems a reasonable hypothesis.

3.3 Components and Roles

The security system we present in this paper is made up of the following elements:

- *Threat Detection Client*: the TD client is in charge of identifying threats and sending reports to the TD server.
- *Threat Detection Server*: the TD server receives the registration requests from sensor nodes. It chooses which sensor(s) will be in monitoring mode for each cluster by taking into account status parameters such as batteries level and available resources. The TD server updates the global network database. It receives the alarms from TD clients and transfers them to the SA Server.
- *Security Adaptation Server*: the SA Server receives the threats from the TD server. It decides then which is the best policy to apply accordingly and stores the new policy for sensor nodes in the security policy mailbox.
- *Security Adaptation Client*: After each boot, the SA Client sends a message to the SA Server to check if there are any policies to be delivered and applies those it receives in return.
- *Security Service*: Various security services are supposed to run on the considered sensor nodes, but two of them are especially relevant. A key management service is assumed to be able to establish security contexts between two nodes. The security level offered by this service translates into various parameters such as algorithms, key lengths or security association life-times. Another security service is the network access control enforcement, which maintains a secure connection between a sensor node and its parent node in the WSN topology. This secure connection is established upon node's entry into the WSN, and remains active as long as the node is part of the WSN. Both of these security services can raise alarms to the local threat detection client even though the node is not actively monitoring its neighborhood, for example if a peer node cannot be authenticated (by other sensor in case of key management, by parent node in case of network access control enforcement).
- *Security Policy mailbox*: a module that stores the generated policies. Then, it delivers them at nodes request. The use of this module is required since the nodes are not synchronized. It also reduces the overall bandwidth consumption.
- *Global Network Database*: contains a global view of the network and the threats detected in the past.

It is populated every time a sensor registers to the TD Server and is updated every time a new policy is applied by the SA Server.

3.4 Operation

This section describes the sequences of actions performed by the client-side and server-side entities. These sequences of actions are categorized according to the sensor mode. This mode can either be bootstrapping, normal mode, or monitoring mode.

Bootstrapping is the mode of a node when it joins the network. The newly joining node is first to send a registration request to the TD server, informing this latter about its potential monitoring abilities. Upon reception of this registration request, the TD server registers the node and sends to the node an acknowledgement (ACK), as well as a configuration message specifying whether the node should remain in normal mode or switch to monitoring mode for a specific amount of time. The decision by the TD server is based on its knowledge of current and, in some cases, foreseen contexts of the candidate monitoring nodes. This contextual information includes data relevant to the nodes resources (e.g. battery level), location, and capabilities (e.g. number of observable neighbors). With this information, the TD server is able to identify the best node in the cluster for acting as a monitoring entity, and to configure it with this role for a certain period of time. Once the monitoring delay expires, the TD server proceeds again to the identification and designation of cluster monitoring node(s).

A node switches to **Monitoring Mode** when ordered to do so by the TD server, either immediately after its bootstrapping, or upon reception of a configuration order, after waking up. The sequence of actions performed when in monitoring node is:

1) Once the TD Client detects a threat, it sends an alarm to the TD server that includes information about the threat. This information contains at least the IP address of the attacker, the IP address(es) of the target(s) and the type of attack;

2) Upon receiving the alarm, the TD Server reports it to the SA server, optionally after having aggregated multiple alarm messages and/or having assessed the quality of the evaluator. Next, it stores the new policies in the security policy mailbox, in order to have them be delivered to the respective SA Clients that will have to enforce them;

3) In monitoring mode, the TD client on the sensor regularly polls the security policy mailbox by sending a dedicated inquiry message to the SA server;

4) The SA server sends the requested policy if it exists. Otherwise, it replies with a message telling the node that it is not to enforce a new policy. Along with security policies, the monitoring node might be instructed to revert back to normal mode if another monitoring node is being designated within the cluster;

5) If a new policy is received, the SA Client enforces it by configuring the security services in accordance with the received policy and acknowledges it through an ACK message;

6) The SA Server receives the ACK and updates the global network database accordingly.

The monitoring mode sequence of actions is represented in Figure 1.

If a malicious node refuses to be placed as a monitoring node, the server could detect that no reports are received from its side and then it could be penalized or excluded from the network. If this node accepts the monitoring process during a period of time but lets other colluding nodes execute attacks while sending positive observations on their behalf. The server can compare its false reports with feedbacks received from other monitoring nodes selected for the same cluster in different time intervals. Hence, it will conclude its malicious behavior as a monitoring node and take the appropriate punishment decision against it.

The **Normal Mode** is the default mode for a bootstrapped sensor node that has not been designated as a monitoring node. In this mode, sensor nodes alternate between active and sleeping states. Upon leaving sleeping state, the node interrogates the SA server about an eventual new policy to enforce. Meanwhile, the node in normal mode may also be instructed to switch into monitoring mode, if required from the evaluation of different nodes energy levels within the considered cluster. The node then performs the task(s) for which it has left the sleeping state. An alarm may be raised by a node in Normal Mode only if one of the run tasks detects a threat. This task would then notify the TD Client through an API call.

The sequence of actions corresponding to normal mode is represented in Figure 2.

3.5 Implementation Environment

The hybrid threat detection solution presented in this paper was implemented using embedded C for sensor nodes, with Atmel studio 6 for AVR/ARM and C++ for the server side. We used Dresden Elektronik sensor devices [5] that have the following specification:

- The gateways are equipped with ARM processor, Ethernet port, power over Ethernet, USB/Serial and IEEE 802.15.4 2.4GHz.

- The sensor devices are equipped with AVR Processor, battery, USB/Serial interface and IEEE 802.15.4 2.4GHz.

Both the gateway and the sensor node use 6LOwPAN protocols. The sensor node turns to the Monitor Mode

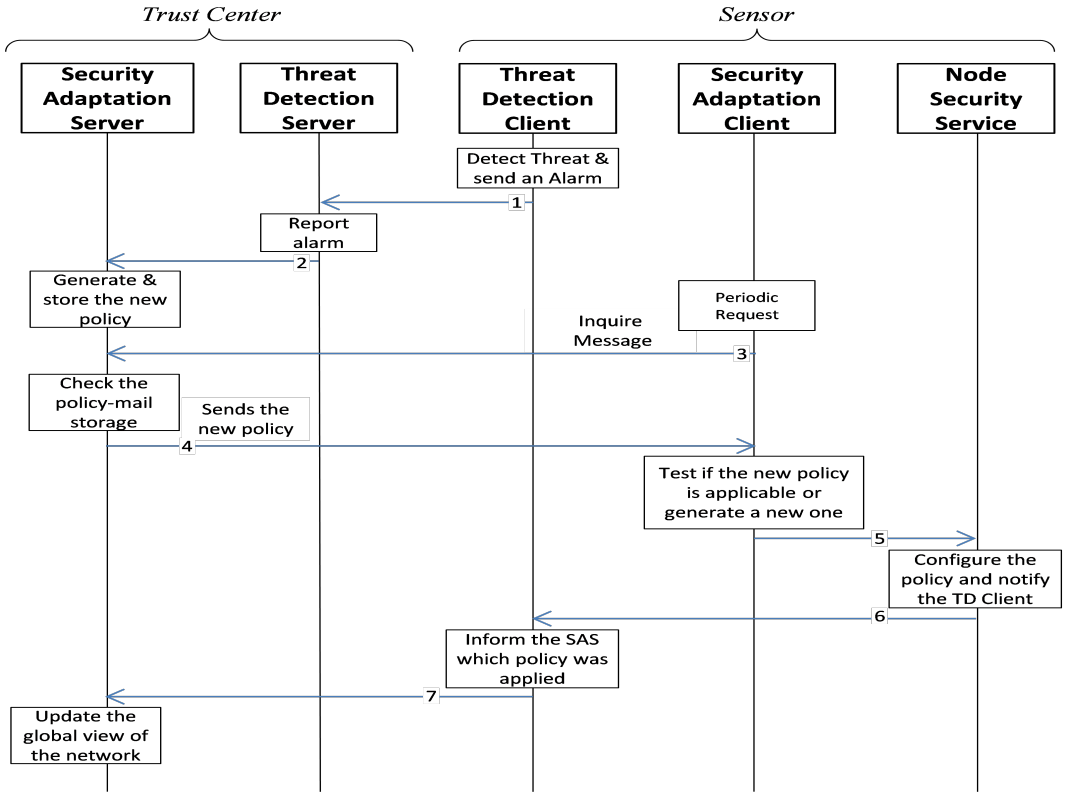


Figure 1: Threat detection and security adaptation in monitoring mode

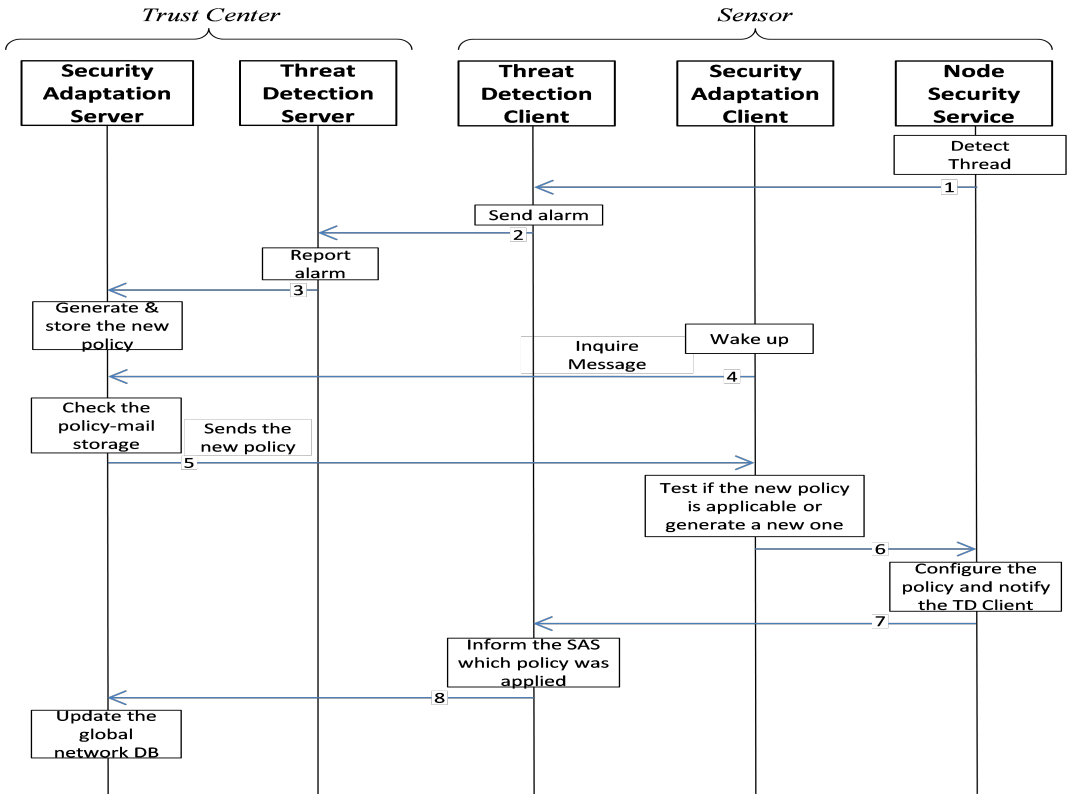


Figure 2: Threat detection and security adaptation in normal mode

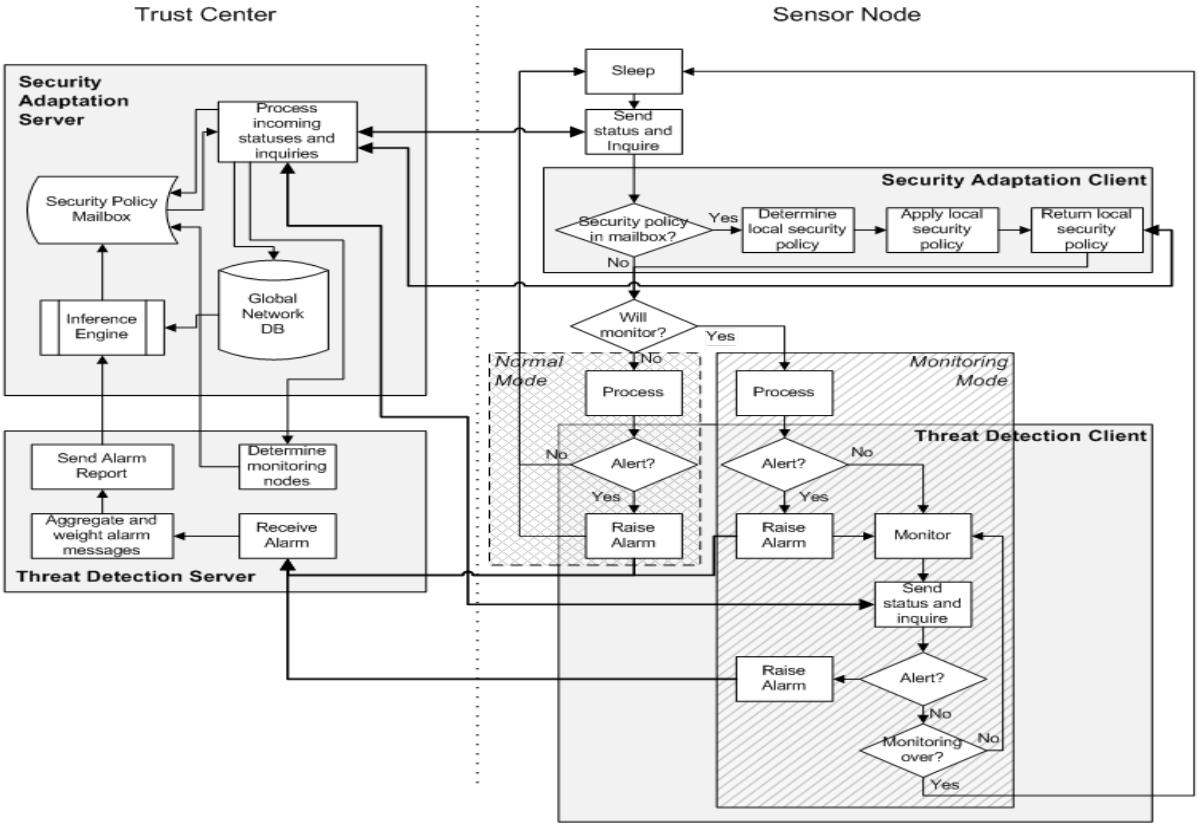


Figure 3: Overall logical architecture and state machines

by switching its 802.15.4 interface to promiscuous mode and disabling its sleep behavior. The overall process of our solution, illustrating its state machines and internal/network message exchanges, is depicted in Figure 3.

4 Performance Evaluation

4.1 Performance Evaluation Criteria

In order to assess the performance of the proposed hybrid threat detection system, it is worth detailing first all criteria that should be taken into account for the evaluation of a threat detection system. Indeed, a threat detection system can be characterized with:

1) The *proportion of false positives* (benevolent behaviors wrongly identified as attacks) and that of false negatives (malicious behaviors not identified as such), as compared with the successful attacks identifications. These ratios and the corresponding performance metrics assume the existence of a list of detectable attacks $\{A_1, \ldots, A_n\}$. Accordingly, an attack A' not detectable - and thus not detected - by the system would not be qualified as a false negative.

2) The *exhaustivity of the detectable attacks list* constitutes a second evaluation metric. Ideally, this list must be built in accordance with a risk analysis carried out on the monitored system, so that the most frequent and/or critical attacks be detectable by the threat detection system.

3) The *reactivity of the threat detection system* that is, the time needed to identify an attacker as malicious. Finally, constrained environments characterized with nodes of low capabilities in terms of computing power and energy capacity exhibit additional requirements, to which a fourth evaluation criterion corresponds.

4) The *energy needed* to locally (at constrained nodes side) operate the threat detection system. This energy cost must be limited, so that the constant use of this system will not consume a high amount of nodes energy re-sources prohibitively fast.

The proposed solution for threat detection does not focus on mitigating false positives and false negatives. From this viewpoint, the local detection algorithm is not improved as compared to the state of the art solutions. Likewise, the weighting and aggregation of alerts at the server side is recommended, but a novel method for doing so is not part of the current proposal. Hence, our proposed system will not be evaluated according to the first evaluation criteria. Our proposed system does also not extend the list of detectable threats, as defined by the second evaluation criteria. The completeness of this list is seen

as orthogonal to the current solution. We acknowledge, however, the high relevance of this problem, especially when put aside the (often neglected) nodes constraints in terms of memory capacity. Indeed, an exhaustive list of detectable attacks would require a corresponding memory space for storing attacks signatures, which would not be affordable to highly-constrained nodes.

While criteria (a) and (b) are thus not considered for assessing the performance of our proposed threat detection system, the importance we put in the reduction of energy consumption leads us to favoring criteria (d) as the most important one for doing so. Meanwhile, we will use the criteria (c) both to guarantee a proper behavior of the proposed threat detection system even though it is instantiated on highly-constrained nodes and to provide a second metric to compare our solution to those proposed in the literature.

4.2 Threat Detection Simulator

We evaluated the performance of our proposed threat detection system by means of a purposely designed discrete time simulator. This section first reviews the generic design decisions that were applied for conceiving this tool. It then considers the models that were used to implement on the simulator both the proposed system and the solution proposed in [6], which is one of the most studied threat detection system for WSNs, to which we intend to compare our work.

The proposed simulation program Csimu (a lightweight discrete time C-based network simulator) starts with initializing a WSN-like topology made of clusters and cluster heads. It also initializes the threat detection system, in particular by assigning per-cluster initial monitoring nodes. We conceived our simulator as a program built around a main loop, with each loop execution corresponding to an elapsed (configurable) time step. A discrete time approach is therefore adopted in this tool. The main loop is made of the following operations:

- Storage of logged values including: overall energy spent, overall energy spent for all threat detection related operations, nodes compromise status.
- Per-node individual processing loop. This second loop is the most important operation. It goes through each node part of the generated topology and performs relevant nodes actions. These consist in:
- Storing per-node values relative to energy spent and energy spent for threat detection operations.
- Updating node energy level in accordance with node status during the last time step. For example, a node that spent the last time step in 'Listen' mode will have its battery level decreased with an energy $E = \Delta t_{Timestep} \times P_{Listen}$, with $\Delta t_{Timestep}$ being a configurable parameter and P_{Listen} the power for the considered node type in listening mode.
- Carrying out node tasks, e.g. uploading a measured value to a remote sink node and updating node energy level accordingly.
- Carrying out attacks, in case the node is compromised. Two types of attacks are represented: A_{DoS} is an example of Denial of Service attack and A_{Comp} represents a compromise attack. Each attack involves the following steps: determination of whether an attack is to occur, choice of the to-be-attacked victim, energy consequences on the attacking node (corresponding to the number of messages to send/receive and the listening durations involved by the attack), energy consequences on the victim, other outcomes on the victim (the A_{DoS} attack has a chance to make the victim compromised), attack detection. The attack detection involves all nodes that are in position to detect the launched attacks, taking in considerations their respective locations (only neighbor nodes are susceptible to discover the attack) and modes (the probability of detecting the attack is much higher for the nodes that are in monitoring mode). In turn, when a node detects an attack, it may either be able to pinpoint the attacker, or may just notice that an attacker is likely to be present in the cluster. In accordance with the detection results and the implemented threat detection system, various actions are to be taken that are detailed in the next section.
- Threat detection system update (detailed in the next subsection) that includes the exclusion of the identified compromised nodes and, in random (non-scripted) mode, a slight probability for each node to become compromised.

4.3 Evaluated Threat Detection Systems

4.3.1 "All Monitor" and "Cluster Monitoring Node Election" Systems

As explained in the related work section, the "all monitor" approach considers that each sensor device acts as an independent monitoring node. Continuing in listening mode, each node tracks its neighbor's communication in order to identify possible attacks. The "Cluster monitoring node election" approach as proposed in [6] relies on monitoring by an elected node, with election process occurring periodically. In accordance with the protocol description in [6], we simulated the election process as a recurring procedure where:

- Each node has its energy level decreased by a decrement corresponding to: (1) a random number generation - we assumed a hash-based random number generation; (2) the computation of a hash on the computed random number and the node identifier; (3) the sending of a message containing said identifier and hash; (4) the waiting for receiving all messages from neighbor (same-cluster) nodes; (5) the actual reception of all of these (n-1) messages (n being the

number of nodes within one cluster); (6) the verification of all received messages through the computation of (n-1) hashes; (7) the operation of the selection function; (8) the sending of a result message.

- As a result of the election, a new node becomes responsible of cluster monitoring. That is, the former monitoring node reverts back to the 'normal' behavior (sleeping mode, with periodic wake up) and the newly elected monitoring node switches to the 'monitoring' behavior (listen mode) for the next period.

4.3.2 Our Hybrid Solution

We implemented our proposed approach on the Csimu simulator by adding within the server processing operation a phase where the threat detection server periodically designates a new monitoring node within all clusters. To that aim, the server maintains a view of the network nodes statuses, which makes it able to designate for each cluster the node with the highest energy level as the new monitoring node. The process happens thus as follows:

- Upon waking up, each node sends to the threat detection server a message prompting for orders and informing the server about its current battery level. The server answers accordingly. The energy costs corresponding to message sending and response waiting, receiving and processing are taken into account and decremented from the nodes energy levels.

- Periodically, the server triggers a new designation of a cluster monitoring responsible node. This does not incur any immediate action. However, the subsequent request for orders of the former monitoring node and the newly designated monitoring node will make these learn, respectively, their orders to switch into 'normal' and 'monitoring' modes.

- The protocol that was implemented in the simulator introduced a slight variation as compared to the approach described above: each cluster is initialized with no active monitoring node. It is assumed that nodes in normal mode can pinpoint - even with much higher probabilities of false positives and false negatives - malicious behaviors and consequently warns the threat detection server, which will in turn mark the cluster as suspicious and assign a monitoring node within its population. The metrics for identifying clusters (proposed approach) / nodes (both approaches) as suspicious / compromised are explained in next section.

4.4 Simulation Parameters: Assumptions and Devices Characteristics

This section reviews the simulation parameters that are used in the simulation environment to rate nodes behaviors, on one hand, and to assess energy consumption, on the other hand.

4.4.1 Assessment of Attack Probability

Without implementing any actual attack detection scheme, we based our simulated threat detection function on a probabilistic model of attack detection, where each behavior (A_1, ..., A_n attacks, as well as the 'A_0' benevolence) can lead to a detection of any behavior within the (A_0,

A_1, ..., A_n) population, with different probabilities. This led us to define an attack recognition matrix M_{AR} as:

$$M_{AR} = \begin{pmatrix} p(D_{A_0}|A_0) & p(D_{A_1}|A_0) & \cdots & p(D_{A_n}|A_0) \\ p(D_{A_0}|A_1) & p(D_{A_1}|A_1) & \cdots & p(D_{A_n}|A_1) \\ \vdots & \vdots & \vdots & \vdots \\ p(D_{A_0}|A_n) & p(D_{A_1}|A_n) & \cdots & p(D_{A_n}|A_n) \end{pmatrix}$$

where $p(D_X|Y)$ corresponds to the probability to detect the event X knowing that event Y occurred.

In fact, two such matrices were defined corresponding to the two modes in which a node can be, respectively normal mode (non-monitoring node involved in usual node operations only) and monitoring node. The probabilities were tuned accordingly, to reflect the fact that a node in monitoring mode is much more accurate in its identifications, whereas the probabilities of false positives ($p(D_{Ai}|\ A_0) > 0$ for some i>0) and false negatives ($p(D_{A0}\ |\ A_i) > 0$ for some i>0)are much higher for a node in normal mode.

Based on this matrix, we were able to compute the probability of actual attack (whatever the A_j, j$\in$ [1; n]) occurrence upon the detection D_{Ai} of a given event A_i.

$$\begin{aligned} p(Attack|D_{A_i}) &= \frac{p(D_{A_i} \cap Attack)}{p(D_{A_i})} \\ &= \frac{p\left(D_{A_i} \cap (A_1 \cup \cdots \cup A_n)\right)}{p(D_{A_i})} \\ &= \frac{\sum_{k=1}^{n} \left(p\left(D_{A_i} A_k\right).p(A_k)\right)}{\sum_{k=0}^{n} \left(p\left(D_{A_i} A_k\right).p(A_k)\right)} \\ &= \frac{\sum_{k=1}^{n} \left(M_{AR}\left[k,\ i\right].p(A_k)\right)}{\sum_{k=0}^{n} \left(M_{AR}\left[k,\ i\right].p(A_k)\right)} \end{aligned}$$

In this computation, the elements M_{AR}[k, i] are the coefficients of the MR matrix defined above. The $p(A_0)$, $p(A_1)$, ..., $p(A_n)$ probabilities of occurrence of events A_0, A_1, ..., A_n are however other parameters that are required, and for which only approximate values can be used. In the simulator we used values derived from the probability of node compromise, and, for a compromised node, the probability of launching a type$i$ attack A_i. In actual conditions, where these probabilities of attacks cannot be precisely known, the system should use cognitive behavior where the optimal values of $p(A_k)$ would be statistically approached from the history of past detected events.

With an attack detection being mapped to the probability of actual attack occurrence, we were able to increase node compromise scores by a proportional coefficient. Eventually, we defined four thresholds t_1, t_2, t_3, and

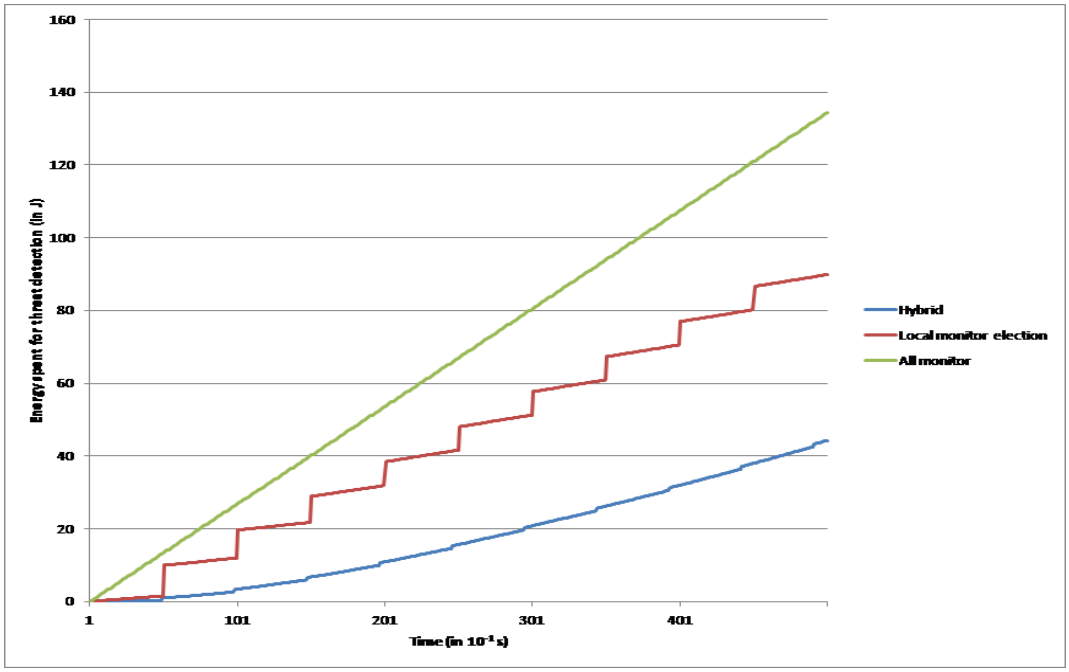


Figure 4: Energy consumption for threat detection operation in all three evaluated systems

t_4 respectively for marking a cluster as 'to be monitored' (when the cluster score exceeds t_1),marking a cluster as 'no longer to be monitored' (when the cluster score falls below t_2), marking a node as 'likely compromised' (when the node score exceeds t_3) and marking a node as 'likely benevolent' (when the node score falls below t_4[1]). We made the cluster and node scores decremented at each execution of the main loop, to mitigate false positives and let the system revert back to normal state after a compromised node has been excluded.

4.4.2 From Node Characteristics to Computation Parameters

We used the node characteristics listed in [11] corresponding to the following parameters: power in transmit mode (54 mW), power in receive mode (61 mW), power in listen mode (60 mW) and energy consumption for one processor cycle (8.64 nJ / cycle). Likewise, sensor nodes throughput was supposed to be equal to 75 kbps, as determined in [11].

Random number generation and hash function required in [6] were assumed to be based on the SHA-1 hash function, for which the required number of cycles per byte was computed, based on eBASH [2], to be 53.24 on the considered processor for a less than 64-byte long message (the shorter the message, the higher the cost for hashing one byte, due to the SHA-1 fixed operations that do not depend on message length).

[1]We did not use that fourth threshold, though in our simulation: instead, we considered that a compromised node was to be removed from the system; and could optionally be re-introduced after a while - but considered as an entirely new node.

4.5 Simulation Results

This section reviews the obtained results in terms of energy cost and efficiency, which prove that our proposed system, though slightly slower to react to malicious operations, is more energy-efficient than the other evaluated solutions.

4.5.1 Energy Consumption

Figure 4 presents the overall energy consumption during all processes related to threat detection for a 5-cluster, 50-node topology. This energy consumption is expressed in Joules per $10^{-}1$ s for the three evaluated systems, namely 'All monitor' (all cluster nodes stay in monitoring mode), 'Elect' (there is only one monitoring node per cluster, which is regularly refreshed through an election carried out between the cluster members [6]) and our proposed 'Hybrid' scheme.

As was expected, the 'All monitor' approach is the most requiring in terms of energy with the 50 nodes (actually, 45 sensor nodes, since energy consumption is not measured on gateways by the simulation environment) consuming around 135 J in 50 s (a result that corresponds to the 60 mW that each node consumes in listening).

Figure 4 also highlights the high energy cost of [6], due to its heavy election processes. Even though only one node monitors a cluster at a time, the cost of election phases (the vertical transitions between plateaus in the curve) does not allow an overall energy gain higher than 33%, as compared with the 'All monitor' approach.

In order to highlight the difference between the local election phase of [6] and the server-assisted monitoring

node designation we proposed, we simulated in Figure 4 a synchronous approach of our hybrid approach, where all cluster nodes in all clusters perform mailbox checking (and, if required, mode change) at the same moment. Nevertheless, the vertical transitions are much smaller than those of [6], as could be expected from the lighter requirements exhibited by the mailbox checking, as compared to local election processing. Eventually, our solution performs approximately twice better than [6] and three times better than the 'All monitor' approach, with a final threat detection energy cost of only 44 J for 45 sensing nodes in 50 seconds (around 20W for each node). It has to be noted that the exponential-like allure of the 'Hybrid' system curve corresponds to the tuned behavior of the system, wherein clusters are initially not monitored (and consume almost no energy for threat detection) and start to get monitored only when compromised nodes reveal their malicious activities and are suspected by nodes. This leads to an acceleration of the energy consumption, which quickly reaches a cruising speed, where all clusters are monitored.

4.5.2 Threat Detection Reactivity

The behavior of all three simulated solutions with respect to compromised nodes identification is detailed below in Figure 5, Figure 6 and Figure 7.

Figure 5, Figure 6 and Figure 7 represent the compromise/react graph for each of the three simulated models. The dark, diamond-marked curves represent the number of unidentified compromised nodes within the system. The light, square-marked peaks represent detections of compromised nodes, and correspond therefore to a decrease of the other curve. Note that all three threat detection systems were tested in the same conditions, with compromising of the same nodes being triggered at the same time (scripted, non-random, compromising mode).

From these results, it appears that the 'All Monitor' approach performs the best with respect to speed in threat identification. Cluster election and the proposed hybrid approach, on the other hand, show almost similar results requiring more regularly a few seconds to detect and react to the presence of an attacking node.

4.5.3 Other Benefits of the Proposed Hybrid Approach

A synchronous embodiment of the proposed hybrid approach was depicted in Figure 4, in order to reflect the lower cost induced by the threat detection management phase (mailbox checking and required actions enforcement) as compared with that of the election approach. Beside this explanatory purpose, there is no need however to mandate synchronicity in our proposed solution: the sensor nodes may very well wake up at different times; the operation of the proposed approach would still work exactly the same. We simulated this asynchronous approach by explicitly requiring each sensor node to wake up at different times. We obtained equivalent performance with respect to reactivity to attacks; on the other hand, the resulting threat detection energy consumption curve presented, as expected, a smoother aspect (Figure 8).

Obviously, the support by our proposed approach of such asynchronicity (hence, of multiple sensor nodes that are not in phase regarding their sleep/wake up periods) could not be provided by the election approach, where all nodes within a common cluster must be involved in a synchronous way in the monitoring node election process. This support of sleeping node represents an important advantage of our proposed approach.

Finally, another benefit of our hybrid approach lies in the fact that interactions with a server entity can be exploited to further improve the threat detection system, by making it dynamically able to react to new threats. Indeed, the orders from the threat detection server to its clients could be extended to carry new attack signatures, thereby making the system both more evolutive and more in line with a given threat model. Simple extensions to the threat detection system, in line with the adaptive security approach, could lead to dynamically choosing which node(s) in a given cluster receive which attack signature(s).

5 Conclusion

This paper presents a threat detection system for industrial wireless sensor networks that could be qualified as hybrid in that it involves a semi-centralized process. The switch from normal threat detection mode to monitoring mode is triggered by the threat detection server, which bases on regular reports from nodes and updates its decision accordingly. We showed that this system, as reactive as the most studied state of the art solutions with respect to identifying threats, performs better from the viewpoint of energy consumption, saving around 50% of energy. The hybrid approach we propose, characterized by centralized management and local instantiation of threat detection, is also more flexible in terms of extension possibilities, opening up interesting development axes for the future of autonomous security in WSNs.

Acknowledgments

This work was financially supported by the EC under grant agreement FP7-ICT-258280 TWISNet project.

References

[1] F. Bao, R. Chen, M. J. Chang, and J. H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust based routing and intrusion detection," *IEEE Transactions of Network Service*, vol. 9, pp. 169–183, 2012.

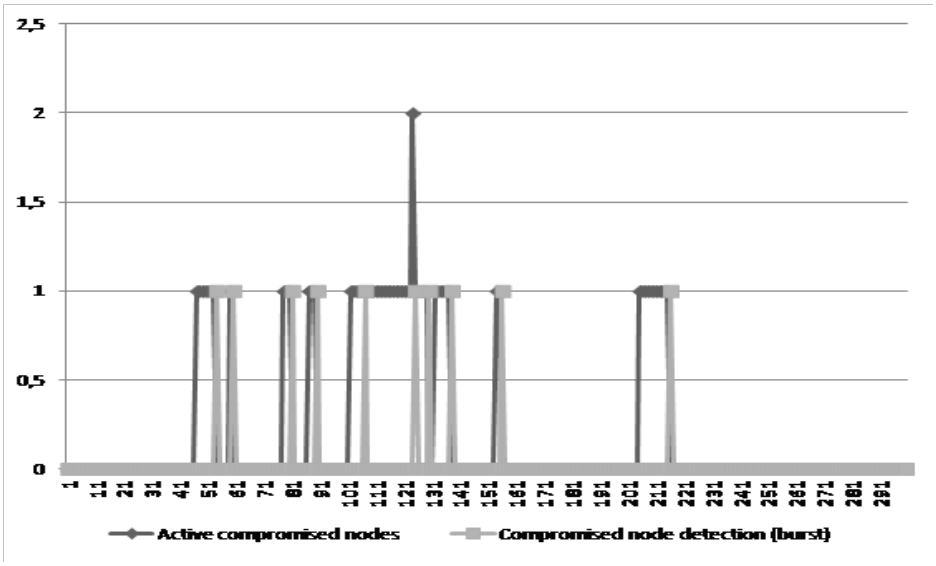


Figure 5: All monitor

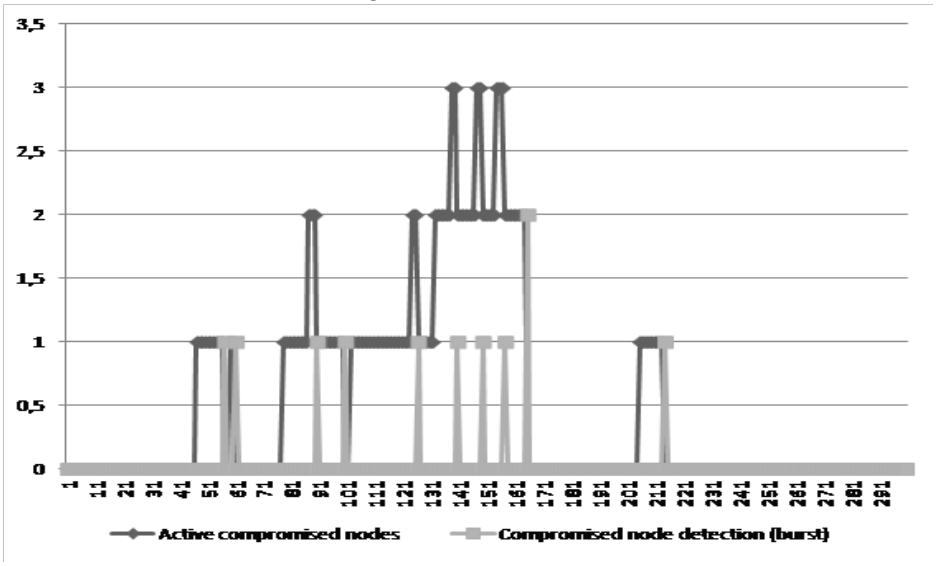


Figure 6: Local monitor election

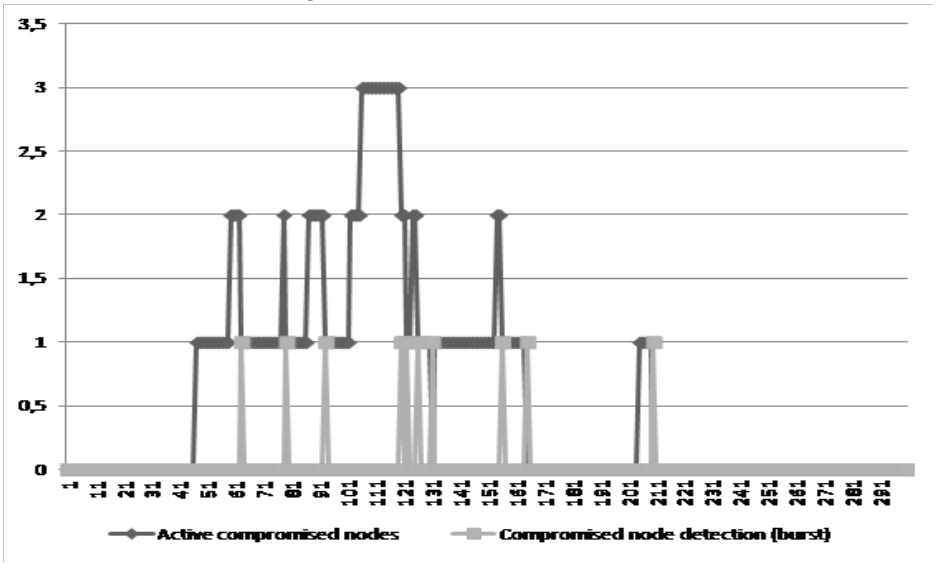


Figure 7: Hybrid approach

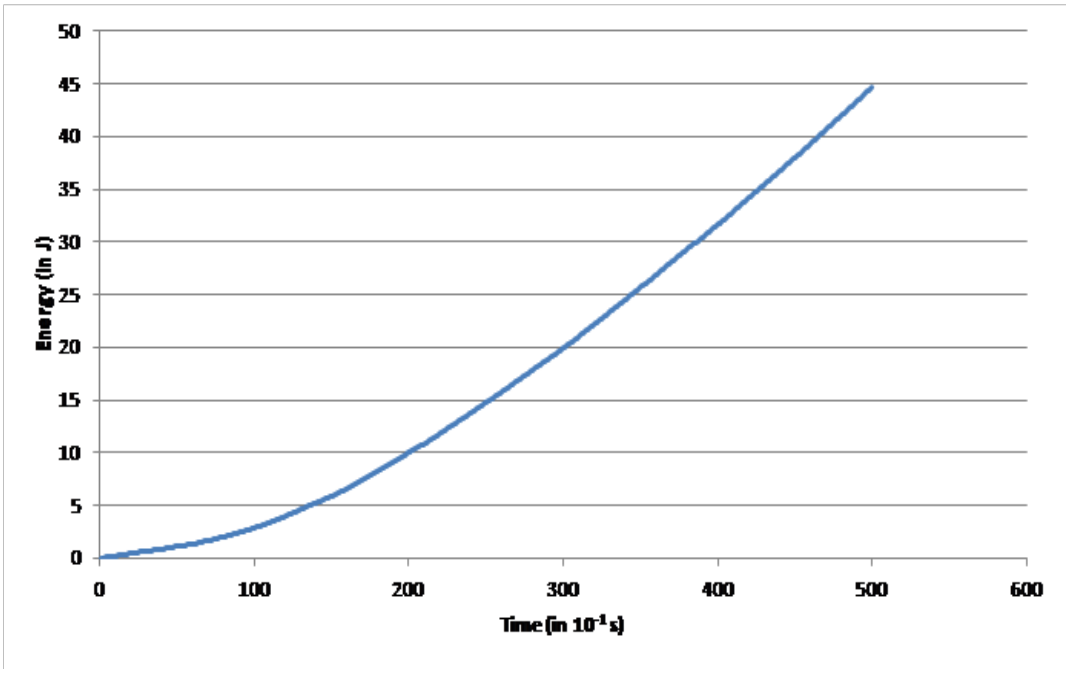


Figure 8: Energy consumption for threat detection operation in hybrid asynchronous mode

[2] D. J. Bernstein, *eBASH: ECRYPT Benchmarking of All Submitted Hashes*, 2013. (`http://www.bench.cr.yp.to/ebash.html`)

[3] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 266–282, 2014.

[4] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of The First ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet'05)*, pp. 16–23, Montreal, Quebec, Canada, Oct. 2005.

[5] D. Elektronik, *Datasheet IEEE 802.15.4 node RCB230 V3.2*, 2015. (`http://www.dresden-elektronik.de`)

[6] Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in *Proceedings of The First ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 135–147, Fairfax, Virginia, Oct. 2003.

[7] K. Ioannis, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in *Proceedings of The 13th European Wireless Conference*, pp. 1–10, Paris, France, Apr. 2007.

[8] I. Krontiris, Z. Benenson, T. Giannetsos, F. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," *Springer Wireless Sensor Networks Journal*, vol. 5432, pp. 263–278, 2009.

[9] I. Krontiris, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in *Proceedings of The 13th European Wireless Conference Computing*, pp. 1–7, Paris, France, Apr. 2007.

[10] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC'06)*, pp. 640–644, Las Vegas, USA, Jan. 2006.

[11] Y. Ben Saied and A. Olivereau, "D-hip: A distributed key exchange scheme for hip-based internet of things," in *Proceedings of The First IEEE WoWMoM Workshop on the Internet of Things: Smart Objects and Services (IoT SoS'12)*, pp. 1–7, San fransisco, USA, June 2012.

[12] R. A. Shaikh, "Group-based trust management scheme for clustered wireless sen-sor networks," *IEEE Transactions on Parallel Distributed Systems*, vol. 20, pp. 1698-1712, 2009.

[13] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public key cryptography for wireless sensor networks," in *Proceedings of The Third IEEE International Conference on Pervasive Computing and Communications*, pp. 324–328, Hawaii, USA, Mar. 2005.

[14] J. Zhang, "A trust management architecture for hierarchical wireless sensor networks," in *Proceedings of The IEEE Conference on Local Computer Networks*, pp. 264–267, Denver, Colorado, Oct. 2010.

Yosra Ben Saied graduated from the University of Pierre et Marie Curie and Telecom SudParis, Paris, France, in 2013 where she obtained her PhD in Telecommunications and Computer Science. In 2014, she joined as a researcher RAMSIS research team at the National

School of Computer Science, Tunisia. Her research activities consist in developing network security solutions for machine-to-machine and cloud environments. She especially focuses on cognitive and collaborative mechanisms.

Alexis Olivereau graduated from Ecole Nationale Supérieure de l'Electronique et ses Applications, Cergy, France in 2000. Between 2000 and 2008 he has been a research engineer in the Motorola Labs research center of Paris, France where he worked on networking security, developing novel protocols for IP-based architectures in the framework of mobile Internet. He participated in various European research projects and earned Motorola "Gold Badge" distinction for his patents filing. He joined the Laboratory of Communicating Systems (LSC) of CEA-LIST in January 2009 as a researcher and is now working on security and privacy aspects of communications in machine-to-machine and cloud environments.

Key Trees Combining Algorithm for Overlapping Resource Access Members

Amar Buchade, Rajesh Ingle
(Corresponding author: Amar Buchade)
Computer Engineering and Information Technology Department, College of Engineering, Pune
Wellesley Rd, Shivajinagar, Pune, Maharashtra 411005, India
(Email: amar.buchade@gmail.com)
(Received July 30, 2015; revised and accepted Sept. 30 & Oct. 13, 2015)

Abstract

In cloud computing environment, resources are accessed by multiple members. Resources may be considered as VM, CPU, Storage etc. Group key management required when multiple members in group accesses the resources securely. In existing group key management, separate key trees are formed even if members are common in another group to access the resources. The solution to this is to form the combined key trees for resources which containing overlapping members. Through the analysis it is observed that computational overhead is decreased by 22% if we combine the key trees than separate key trees for each resource.

Keywords: Group key management, resource, resource tree, security

1 Introduction

In Cloud computing, resources may be simultaneously accessed by multiple members. Here the resources may be considered as database, CPU, storage, applications. The members can be overlapped to access the above resources.

Other example HDTV, users can subscribe to various layers such as base layer, medium layer and enhanced layer channel. Users which subscribed medium layer can watch HDTV base layer as well as medium layer channel. User which subscribed enhanced layer can watch base layer, medium layer as well as enhanced layer channel.

In existing group key management, single separate key tree is built to form group key even if members are overlapped to access the resources. Member has to maintain keys for each key tree. The solution to this is to form the combined key trees for resources which containing common members.

To form the group key, TGDH protocol is used [3, 6, 7]. More specifically our contributions are

1) Combined key tree algorithm;
2) Algorithms: Single, batch join, single, batch leave;
3) Formulation of computational cost;
4) Computation cost analysis of resource key formation for separate key trees and combined key tree in terms of number of modulation exponentiation operations and sequential operations.

The paper is organized as follows. In Section 2, we present about resource key tree. Section 3 presents combining key trees algorithm, Section 4 covers Results and Analysis, Section 5 presents conclusion.

2 Resource

2.1 Initializations

Let Resource group $R = \{R_1, R_2, R_3, R_4, \cdots, R_n\}$. Consider two resources $R1$ and $R2$.

$R1 = \{m_1, m_2, m_3, m_4, \cdots, m_n\}$ be the members accessing resource $R1$.

$R2 = \{n_1, n_2, n_3, n_4, \cdots, n_n\}$ be the members accessing resource $R2$.

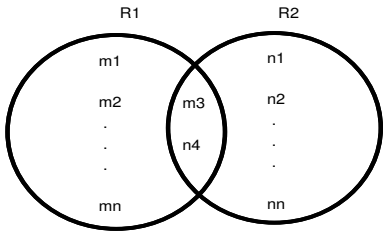


Figure 1: Resources, members representation

From Figure 1 it is observed that m3 and n4 are overlapped to access the Cloud resources. Assume $R1 \cap R2 = cm$, where cm is number of overlapping members which accesses the resources $R1$ and $R2$.

2.2 Resource Tree For Group Key Formation

Figure 2 shows resource tree with leaf nodes represents group members m_1, m_2, m_3, m_4, etc.

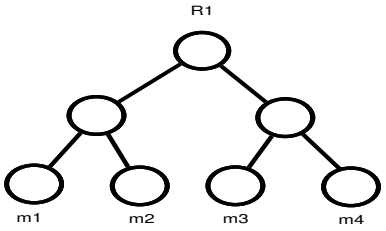


Figure 2: Resource key tree

In TGDH [1, 2, 7, 12, 13], group key is formed from bottom-up fashion. Following are the steps to form the group key.

- Members m_1, m_2, m_3, and m_4 have $\alpha 1, \alpha 2, \alpha 3$ and $\alpha 4$ private keys respectively.
- Each member forms the public key called as blinded key. In this case, g is generator, p is prime number.
- Member blinded keys are $g^{\alpha 1} \bmod p$, $g^{\alpha 2} \bmod p$, $g^{\alpha 3} \bmod p$, $g^{\alpha 4} \bmod p$.
- Each member with its key and sibling blinded key forms intermediate key. For that path from leaf node to root node is traversed.
- Resource group key is formed as below.

$$g^{\alpha 1 \alpha 2 \alpha 3 \alpha 4} \bmod p.$$

Number of modular exponential operations required when four members are equal to 12; In general, when number of members are N, the modular exponential operations are equal to

$$N + N \log_2 N = N(1 + \log_2 N).$$

2.3 Resource Membership Matrix

Every member that accesses the resource, entry is made in resource membership matrix also for any member that joins/leaves in single or batch makes. Resource matrix contains the following entries.

Rows represents members $\{m_1, m_2, m_3, \cdots, m_n\}$ and columns represents resources $\{R_1, R_2, R_3, \cdots, R_n\}$

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots \\ 1 & 1 & \cdots & \cdots \\ 1 & 0 & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \end{bmatrix}$$

It shows that there are $R1$, $R2$, $\cdots$, Rn resources. Member m_1 accesses resource $R1$ while member m_2 accesses resource $R1$ and $R2$, i.e. overlapped to access the resources $R1$ and $R2$.

3 Combining Resource Key Trees Algorithm

In existing key management algorithm [5, 8, 9, 10, 11, 14, 15] seperate key tree is built for each resource, even if same members are accessing multiple resources. Thus we can combine multiple resource key trees. Algorithm 1 illustrates combining resource key trees algorithm. Computation cost analysis is given in Section 4.

Algorithm 1 Combining resource key trees algorithm

1: Begin
2: Let R be the set of resources R_1, R_2, R_3, R_4, $\cdots$, R_n;
3: Let $M = \{m_1, m_2, m_3, \cdots, m_n\}$ be set of members;
4: Each member keep the track of members through resource access matrix;
Rows represents members $\{m_1, m_2, m_3, \cdots, m_n\}$ and columns represents resources $\{R_1, R_2, R_3, \cdots, R_n\}$:

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots \\ 1 & 1 & \cdots & \cdots \\ 1 & 0 & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \end{bmatrix}$$

5: Identify the members which are overlapped to access multiple resources;
6: Build the key tree of overlapped members. Maintain the following entries in Table 1;
7: Identify the members which are not overlapped. Build the key tree of members which are not overlapped;
8: Combine the trees which are formed during Step 6 and Step 7;
9: End

Table 1: Resources containing overlapped members

Index	Resources	Overlapping members
1	$R1, R2$	m_2

3.1 Computational Cost for Group Key Formation

There can be multiple members overlapping to any resources. Let MEO denotes Modular Exponential Operations; RMM denotes Resource Membership Matrix. MEO after combining key trees is equal to MEO for separate key trees minus MEO due to overlapping members:

$$\begin{aligned} & MEO\_for\_separate\_key\_trees \\ = & \sum_{i=1}^{k} N_i(1 + \log_2 N_i) \end{aligned}$$

and

$$MEO\_due\_to\_overlapping\_member = \sum_{index=1}^{Tindex} (RTcount[index] - 1) \cdot (CM[index](1 + \log_2 CM[index]))$$

where

N is equal to number of members per key tree;

k is equal to total number of key trees;

$Tindex$ is equal to number of entries in Table 1;

$RTcount$ is equal to total resource count per entry;

CM is equal to number of members overlapped per entry.

It is observed that computation cost in terms of number of modular exponential for separate key tree is $O(N)$ while for key trees combined is $O(N - RCm)$.

Thus we can observe that number of modular exponential operations required in separate key trees is more, i.e. RCm compared to the combined key trees.

Table 2: Complexity

Best Case	Worst Case
$\Omega(N)$	O(2N)

Table 2 shown the complexity in terms of MEO:

1) Best case complexity in terms of MEO is when all members of resource groups are overlapped to access the resources.

2) Worst case complexity in terms of MEO is when members of resource groups are not overlapped to access the resources.

3.2 Single Member Join

Algorithm 2 explains algorithm for a member single join.

In single member join algorithm, it requires two messages:

1) Message from member for accessing the resource.

2) Message from sponsor to send the blinded key to form the group key.

Table 3 shows the single join. Table 4 represents the message while joining the group for accessing the resource. The member which wants the access of resource broadcast the message containing message id, its originating address, list of resource membership, request for which resource. This helps to each member to make the entry in resource membership matrix.

Algorithm 2 Single member join

1: Begin
2: Joining member broadcasts for resource access with its message contains whether it is already a member of other resource.
3: Each member of resource/s including sponsor notices it and makes this entry in resource membership matrix.
4: Each member looks into resource membership matrix.
5: Each member builds its own key tree by considering overlapping members and non overlapping:
   1) Joining member which is not going to be overlapped, it is added as per TGDH.
   2) Joining member which is already accessing other resource (overlapping) becomes the sponsor.
6: Joining member which is overlapped forms key graph of resources.
7: Non overlapping member maintains its own key tree.
8: Sponsor computes the blinded key and broadcasts it.
9: Group key establishment as in Section 2.2.
10: End

3.3 Batch Join

There can be members which simultaneously access the resources [4]. The following things can be happened,

1) Some members in group may access single resource;

2) Some members in group may access the multiple resources at particular instant of time.

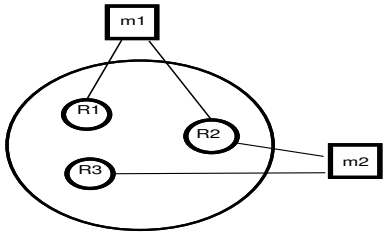


Figure 3: Members join

Figure 3 shows that member m_1 accesses the resources $R1$, $R2$. Member m2 accesses the resources $R2$ and $R3$.

In existing key management, seperate/isolated resource key tree is formed for members that accessing multiple resources. Algorithm 3 explaining when multiple members requests for accessing the resources.

3.3.1 Finding Suitable Position for Common Members in Tree

After building subtree of the overlapping members,it is inserted at the root of the tree to minimize the height of the tree.

Table 3: Single join

Messages	Unicast	Broadcast	MEO
2	0	2	$(N+1)(1+\log_2(N+1))$

Table 4: Message from joining member

Message Id	Source Address	Broadcast Address	Resource Membership	Resource Request

Algorithm 3 Batch join

1: Begin
2: Joining members broadcasts for resource with its message contains whether it is already member of other resource.
3: Each member of resource/s including sponsor notices it and makes the entries in resource membership matrix.
4: Joining members can be categorized as:
   1) Some members newly requesting resource.
   2) Some members already accessing of resources and require to access other resource.
5: Build key graph as per Algorithm 1.
6: Sponsor computes the blinded key and broadcasts it.
7: Group key establishment as in Section 2.2.
8: End

3.3.2 Sponsor Selection for Batch Join

Sponsor is overlapped member which accessing the resources. Otherwise it is selected as TGDH approach [7].

Table 5 shows when members join for the resources access. Here n represents number of members currently added for the resources access.

3.4 Single Leave

Algorithm 4 explains algorithm for a member single leave.

Algorithm 4 Single member leave

1: Begin
2: Leaving member broadcasts that it is leaving from particular resource.
3: Each member of resource/s including sponsor notices it and makes this entry in resource membership matrix.
4: Each member builds its own key tree by considering overlapping members and non overlapping members.
5: Sponsor computes the blinded key and broadcasts it.
6: Group key establishment as in Section 2.2.
7: End

In single member leave algorithm, it requires two messages:

1) Message from member for non-access the resource/s.
2) Message from sponsor to send the blinded key to form the group key.

Table 6 shows the single leave analysis. Table 7 represents the message while leaving from the group. The member which leaves from the group, broadcast the message containing message id, its originating address, non-access of resource. This helps to each member to make the entry in resource membership matrix.

3.5 Batch Leave

There can be members which simultaneously access the resources and completes access of resources. In these members,

1) Member may finishes access of single resource;
2) Members may finishes access of the multiple resources.

Algorithm 5 explains algorithm for members Batch leave. Table 8 represents the batch leave analysis.

Algorithm 5 Batch leave

1: Begin
2: Leaving members broadcasts that it is leaving from particular resource as overlapping members in resource groups or as per TGDH if not overlapping exists.
3: If the leaving member is itself sponsor, sponsor selection.
4: Each member of resource/s including sponsor notices it and makes these entries in resource membership matrix.
5: Build the key graph as per Algorithm 1.
6: Sponsor computes the blinded key and broadcasts it.
7: Group key establishment as in Section 2.2.
8: End

3.5.1 Sponsor Selection in Batch Leave

If the sponsor is leaving member, sponsor is selected as one of the overlapping member. If no overlapping member exists, sponsor is selected as per TGDH [7].

Table 5: Batch join analysis

Messages	Unicast	Broadcast	MEO
2	0	2	$(N+n)(1+\log_2(N+n))$

Table 6: Single leave analysis

Messages	**Unicast**	**Broadcast**	**MEO**
2	0	2	$(N-1)(1+\log_2(N-1))$

Table 7: Message details in single member leave

Message Id	Source Address	Broadcast Address	Non-access of which Resource

Table 8: Batch leave analysis

Messages	**Unicast**	**Broadcast**	**MEO**
2	0	2	$(N-n)(1+\log_2(N-n))$

4 Results and Analysis

Analysis is done by taking resources, varying members size. From Figure 4, it is observed that when No. of resources are 2, Total number of Members=250 and overlapping members in resources are increased, number of exponential operations are decreased (22.41%) when key trees are combined.

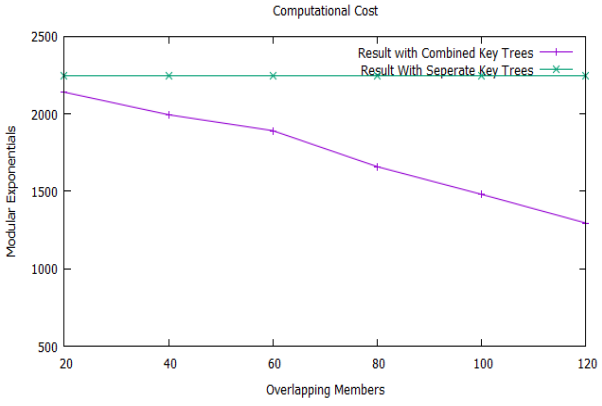


Figure 4: Computational cost, number of resources = 2, total members = 250

From Figure 5, it is observed that average computation cost with separate key trees increased by 21.82% comparing with combining key tree.

From Figure 6, it is observed that when Group size = 100, overlapped members = 10 and as we increase the number of resources, modular exponential operations are decreased when key trees are combined.

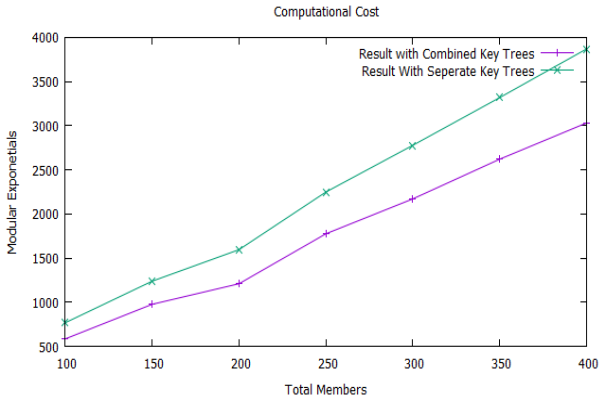


Figure 5: Average computational cost, number of resources = 2

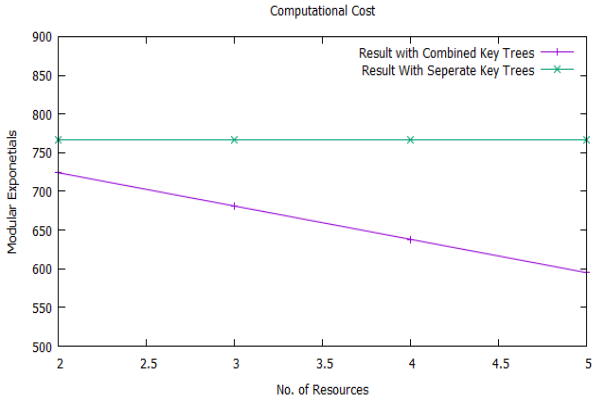


Figure 6: Average computational cost, total members = 100, overlapping members = 10

5 Conclusions

In Cloud computing, resources such as CPU, VM, database, storage and applications are simultaneously accessed by multiple members. The members can be overlapped to access the above resources. Group key is formed by contributing share of each members with TGDH approach. In existing group key management, single separate key tree is built to form group key even if members are overlapped to access the resources.

From the result and analysis, it is observed that there is reduction of computation cost more than 22% when resource key trees are combined.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] X. Gu, Y. Zhao, and J. Yang, "Reducing rekeying time using an integrated group key agreement scheme," *Journal of Communications and Networks*, vol. 14, no. 4, pp. 418–428, 2012.

[3] H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, "Key management for content access control in a hierarchy," *Computer Networks*, vol. 51, no. 11, pp. 3197–3219, 2007.

[4] R. Ingle and G. Sivakumar, "Tunable group key agreement," in *32nd IEEE Conference on Local Computer Networks (LCN'07)*, pp. 1017–1024, 2007.

[5] S. Jarecki, J. Kim, and G. Tsudik, "Flexible robust group key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 879–886, 2011.

[6] D. H. Je, J. S. Lee, Y. Park, and S. W. Seo, "Computation-and-storage-efficient key tree management protocol for secure multicast communications," *Computer Communications*, vol. 33, no. 2, pp. 136–148, 2010.

[7] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 60–96, 2004.

[8] I. Lam, S. Szebeni, and L. Buttyán, "Invitation-oriented tgdh: Key management for dynamic groups in an asynchronous communication model," in *41st IEEE International Conference on Parallel Processing Workshops (ICPPW'12)*, pp. 269–276, 2012.

[9] D. Li and S. Sampalli, "A hybrid group key management protocol for reliable and authenticated rekeying," *International Journal of Network Security*, vol. 6, no. 3, pp. 270–281, 2008.

[10] V. S. Naresh and N. V. E. S. Murthy, "Elliptic curve based dynamic contributory group key agreement protocol for secure group communication over ad-hoc networks," *International Journal of Network Security*, vol. 17, no. 5, pp. 588–596, 2015.

[11] M. Rajaram and D. Thilagavathy, "An interval based contributory key agreement," in *IEEE International Conference on Wireless Communication and Sensor Computing (ICWCSC'10)*, pp. 1–6, 2010.

[12] Y. Sun and K. J. Liu, "Hierarchical group access control for secure multicast communications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1514–1526, 2007.

[13] G. Wang, J. Ouyang, H. H. Chen, and M. Guo, "Efficient group key management for multi-privileged groups," *Computer Communications*, vol. 30, no. 11, pp. 2497–2509, 2007.

[14] H. Xiong, X. Zhang, W. Zhu, and D. Yao, "Cloudseal: End-to-end content protection in cloud-based storage and delivery services," in *Security and Privacy in Communication Networks*, pp. 491–500, Springer, 2012.

[15] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," *CIEEE Transactions on loud Computing*, vol. 2, no. 4, pp. 459–470, 2014.

Amar Buchade Amar Buchade is pursuing Ph.D.(Computer Engineering) from College of Engineering Research Centre, Pune under Savitribai Phule Pune University. He has received B.E. and M.E. in CSE from Walchand College of Engineering, Sangli in 2002 and 2005 respectively. His research area is Distributed System, Cloud computing and Security.

Rajesh Ingle is adjunct Professor at Department of Computer Engineering, Government College of Engineering Pune. He is Professor in Department of Computer Engineering, Pune Institute of Computer Technology, Pune. He has received Ph.D. CSE from Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai. He has received the B.E. Computer Engineering from Savitribai Phule University of Pune, and M.E. Computer Engineering from Government College of Engineering, Savitribai Phule Pune University. He has also received M.S. Software Systems from BITS, Pilani, India. He is a senior member of the IEEE, IEEE Communications Society, and IEEE Computer Society. His research area is Distributed system security, Grid middleware, Cloud security, Multimedia networks and spontaneously networked environments.

Imperceptible Image Authentication Using Wavelets

Anirban Goswami[1] and Nabin Ghoshal[2]
(Corresponding author: Anirban Goswami)
Department of Information Technology, Techno India[1]
EM-4/1, Sector-V, Salt Lake, Kolkata, West Bengal 700091, India
Department of Engineering and Technological Studies, University of Kalyani[2]
Kalyani, Nadia-741235, India
(Email: angos.kol@gmail.com)
(Received May 25, 2015; revised and accepted Sept. 6 & Oct. 4, 2015)

Abstract

This paper introduces an enhanced image authentication technique providing greater security with uncompromising visual quality. To augment data security, the authenticating information is diffused into the transformed coefficients of both the levels after a two level Discrete Haar Wavelet Transform. In addition a bit level noise reduction algorithm increases the imperceptibility of the added noise. The extraction algorithm is completely blind and authenticity is verified by regenerating a message digest at the receiving end. The algorithm has been tested against some related attacks and is appropriate in smart card design. Performance comparisons exhibit significant growth over other similar techniques.

Keywords: Copy attack, DWT, image authentication, MD, SSIM

1 Introduction

In present scenario, extensive use of internet facility in daily activities has indulged in certain issues like ownership of digital images, authenticity of ownership claims, copyright, data integrity, fraud detection, self-correcting images etc. So, covert communication is gaining importance and data (Image/Text/Audio/Video) hiding within a cover image has become an important factor.

The issues that are involved in **data hiding** are: 1) Perceptibility: Embedding of secret information in a cover medium with visually acceptable distortion level, 2) Capacity: Change in the volume of secret data with respect to perceptibility and 3) Robustness: Resistance against effort to destroy, remove, or change the embedded data. Various data-hiding schemes are available which emphasize on hiding high amount of secret data within a cover image without destroying the aspect of imperceptibility.

The boom in the internet technology has resulted in more and more digital images getting transmitted over non-secure channels very quickly. So, military, medical and quality control images must be protected against unauthorized manipulation during transmission. Moreover, due to unavoidable interference during transmission original secret data may not reach an intended receiver. This run time problem is taken care of by the process of **image authentication**, i.e. the secret data is hidden within a cover image in an imperceptible manner which can only be deciphered by an intended receiver.

To protect the authenticity of the documents, several approaches including cryptography, watermarking, digital signatures and steganography based on the image content are proposed. The concept of cryptography (encryption and decryption algorithms) was used to protect the secrecy of the message and its communication. But slowly the concept of cryptography became very weak and the secrecy of the existence of the data became a point of concern. So, **watermarking** was introduced to hide digital information in a carrier signal without indulging any special curiosity for the attackers. Digital watermarking facilitates users to handle a secret document legally along with the necessary security. A further refinement, i.e. invisible watermarking confirms that an authorized person is only eligible to extract a watermark utilizing some mathematical calculations. This defines more security and robustness than visible watermarking in the domain of data privacy.

Moreover, digital image steganography [4] based image authentication plays a vital role in preserving and protecting secured documents by showing effective resilience against attempts to corrupt the hidden data. So, every algorithm in this domain must consider certain factors like 1) perceptual transparency, i.e. degradation in the quality of the cover image is insignificant and 2) the volume of payload data [20] and robustness of embedding, i.e. resistance against related attacks namely AWGN, filtering, lossy compression, scaling and cropping.

The key issue, i.e. effectiveness of robust embedding [19], being highly dependent on the pattern of concealment, appropriate domain (Spatial domain [11, 21, 36, 39] or Transform domain [8, 15, 17]) of the cover image and the position of embedding are major concerns. In contrast to spatial domain, choosing spectral domain of an image for embedding proves more credible. Amongst the available transform domain techniques viz. Discrete Cosine transforms (DCT), Discrete Fourier transforms (DFT) [18], Z transforms [16] etc., recently algorithms are focusing more on Discrete Wavelet transforms (DWT) [24, 31] for its two exclusive features namely Multi-resolution analysis (MRA) and rectification of the problem of time - frequency resolution as mentioned in the theoretical aspects of HVS [22].

Some of the existing algorithms are discussed in this context. Embedding of secret messages in high frequency coefficients and utilizing the unchanged low frequency coefficients for improving the image quality was proposed by Chen et al. in "A DWT Based Approach for Image Steganography" [7]. A lossy image compression using wavelet technique was implemented by Raviraj et al. in "The Modified 2D-Haar Wavelet Transformation in Image Compression" [29] where different compression thresholds for the wavelet coefficients was considered to improve the quality of the reconstructed image. Similarly, a lossy image compression method was also proposed by Tamboli et al. in "Image Compression using Haar Wavelet Transform" [32] where different related compression thresholds were applied to minimize the computational requirements. In "Robust Digital Image Steganography within Coefficient Difference on Integer Haar Wavelet Transform" [1], Abu et al. tried to embed the secret message in the difference values of two adjacent 1-level Integer Haar Wavelet transformed coefficients. 1 level of resolution using Discrete Haar Wavelet transform and embedding in all the four coefficients was also proposed by Bhattacharyya et al. in "Data Hiding in Images in Discrete Wavelet Domain Using PMM" [3]. In another algorithm "DWT Based Watermarking Algorithm using Haar Wavelet" [2], Anuradha et al. also suggested embedding into 1 level decomposed coefficients. Vanitha et al. in "A Review on Steganography - Least Significant Bit Algorithm and Discrete Wavelet Transform Algorithm" [34] considered only the LSB position of Discrete Haar Wavelet transformed coefficients to embed the message bits. In another algorithm "Implementation of Image Steganography using 2-Level DWT Technique" [35], Verma et al. tried to insert the secret data in the LL sub-band of the transformed coefficients.

From the above facts, it appears that some of the algorithms were developed using lossy wavelet based compression technique [30], whereas in others either there is a use of only 1- level Haar resolution or there is degradation in the image quality after the embedding of secret data. In the proposed algorithm, a high image compression ratio is maintained using lossless image compression [30] technique. Moreover, the cover image is decomposed to 2-level of resolution and both the levels are simultaneously used for embedding. In contrast to normal LSB position for embedding [33], the proposed algorithm uses the pseudo random nature of embedding position to firmly authenticate the resilience against attempt to corrupt the data by an intruder. Moreover a decent image quality is maintained by successfully decreasing the difference between the cover and stego image in spite of embedding at different dynamic LSB positions.

The theory of Discrete Haar Wavelet transform (DHWT) and Inverse Discrete Haar Wavelet Transform (IDHWT) are discussed in the next section.

2 Concept of DHWT and IDHWT

DHWT considers both low pass and high pass filters to extract the low frequency (approximation) coefficients and high frequency (detail) coefficients of a signal [10] respectively. In $n \times n$ matrix these filters are applied along the rows and then along the columns at every level of decomposition. In the first level the generated sub-bands are

1) LL (low-low frequency) representing approximation band;
2) LH (low - high frequency) representing vertical band;
3) HL (high - low frequency) representing horizontal band;
4) HH (high - high frequency) representing diagonal band.

Each successive decomposition level further, utilizes the LL sub-band of the previous level.

Mathematically, DHWT replaces a sequence of values by its pair wise average $x_{n-1,i}$ and difference $d_{n-1,i}$ values calculated as in Equation (1):

$$\left\{ \begin{array}{l} x_{n-1,i} = (x_{n,2i} + x_{n,2i+1})/2 \\ d_{n-1,i} = (x_{n,2i} - x_{n,2i+1})/2 \end{array} \right\} \quad (1)$$

For example,

1) As consecutive pairs of input sequences having the first element as even index are used for calculating the averages and differences, the total number of elements in each set, i.e. $(x_{n-1,i})$ and $(d_{n-1,i})$ is exactly equal to half the number of elements mentioned in the original sequence.
2) The two sequences $(x_{n-1,i})$ and $(d_{n-1,i})$ are concatenated to generate a new sequence of similar length as that of the input sequence. For example if the original sequence is (10, 13, 25, 26, 29, 21, 7, 15), then the resulting sequence will be $(11.5, 25.5, 25, 11, -1.5, -0.5, 4, -4)$. This sequence can be visualized as 2 halves:
    a. Averages from the original sequence, i.e. a coarser approximation to the original signal is considered as the first half;

$$M_0 = \begin{bmatrix} a(191) & b(187) \\ c(171) & d(151) \end{bmatrix} \xrightarrow{\text{After DHWT}} M_1 = \begin{bmatrix} a'(175) & b'(6) \\ c'(14) & d'(-4) \end{bmatrix} \xrightarrow{\text{After IDHWT}} M_2 = \begin{bmatrix} a(191) & b(187) \\ c(171) & d(151) \end{bmatrix}$$

Figure 1: An example

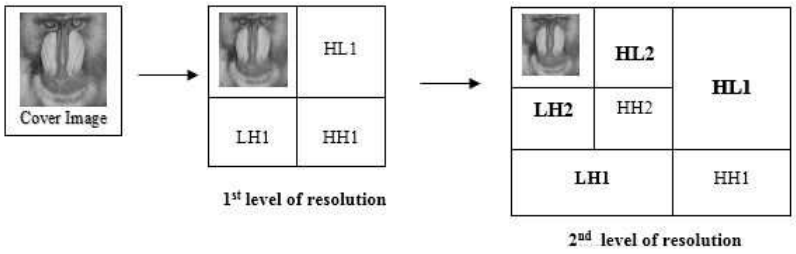


Figure 2: Two-level of resolution using discrete Haar wavelet transform

b. The second half contain the details or approximation errors of the first half.

These transformations do not increase the volume of data. In reverse process, Inverse Discrete Haar Wavelet Transform (IDWT) reconstructs the original sequence by using Equation (2):

$$\left\{ \begin{array}{l} x_{n,2i} = (x_{n-1,i} + d_{n-1,i}) \\ x_{n,2i+1} = (x_{n-1,i} - d_{n-1,i}) \end{array} \right\} \quad (2)$$

In both these equations n represents the total number of elements in a set and i denote a particular position within the set.

In practical application, the implementation of DHWT and IDHWT on a 2 x 2 matrix is explained in Figure 1.

Here M_0 is the original spatial matrix, M_1 is the transformed matrix and M_2 is the reformed spatial matrix.

The transformed coefficients are generated by using

$$\begin{aligned} a' &= ((a+b)+(c+d))/4, \\ b' &= ((a-b)+(c-d))/4, \\ c' &= ((a+b)-(c+d))/4 \\ d' &= ((a-b)-(c-d))/4. \end{aligned}$$

The spatial components are recalculated as

$$\begin{aligned} a &= ((a'+b')+(c'+d')), \\ b &= ((a'-b')+(c'-d')), \\ c &= ((a'+b')-(c'+d')) \\ d &= ((a'-b')-(c'-d')). \end{aligned}$$

Here a, b, c and d are spatial components and a', b', c' and d' are their frequency counterparts.

So, the effectiveness of DHWT refers to:

1) Creation of sub images at multiple resolutions which is similar to a process of HVS [13, 14, 38];

2) The averaging and differencing operations at multiple resolutions is similar to some important image analyzing methods namely Laplacian pyramid method of Burt et al. [5] and the Mumford-Shah theorem [26];

3) Decent correlation between fractal theory [9] and wavelet transforms. The procedures for effective hiding and proper extraction of the secret data are explained in the next section.

3 The Technique

In the embedding technique, the cover image is considered as a set of non-overlapping mask each of size 4×4. 2D DHWT is applied on each of these mask to obtain frequency coefficients and the decomposition is done up to 2 levels. The payload is embedded in the middle frequency bands and three areas of embedding viz.

1) The coefficients of HL2, LH2 and 4 coefficients of HL1;

2) The coefficients of HL2, LH2 and 4 coefficients of LH1;

3) 4 coefficients of LH1 and 4 coefficients of HL1 are proposed. The areas are highlighted in Figure 2.

The formation of stego coefficients can be mathematically expressed as: $s'(m,n) = s(m,n) + \alpha \times w(k)$, where $s(m,n)$ is the host signal, $s'(m,n)$ is the stego signal, $w(k)$ is the payload in form of a distributed sequence and α is the scaling factor to ascertain the strength of the payload signal. The value of α is controlled to maintain a coordination between the imperceptibility and robustness of embedding. The bitwise payload sequence is generated from 1) payload size of 32 bit which represents the combined size of image header and image data, 2) payload message digest (160 bit) and 3) payload data.

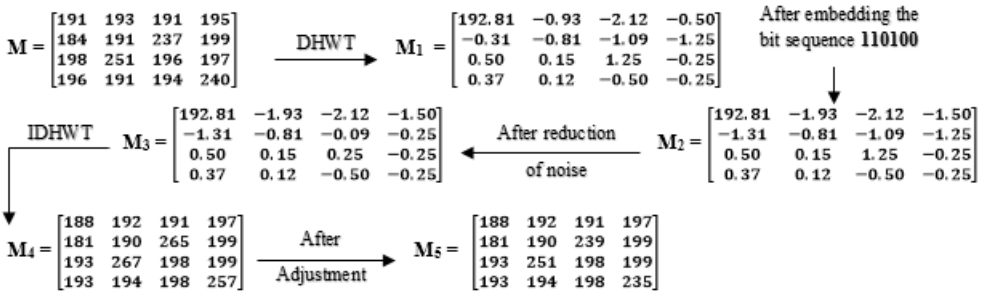


Figure 3: Embedding

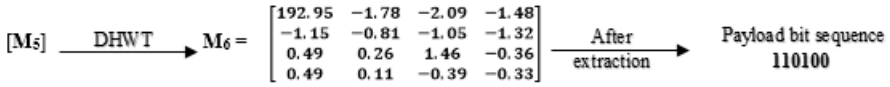


Figure 4: Extraction

An example of the embedding and extraction procedure are shown in Figure 3 and Figure 4 respectively.

In the embedding process, M represents a set of spatial values. On application of DHWT, M1 is generated which contains a set of corresponding frequency values. A bit sequence 110100 is fabricated at pseudo-random generated positions (Subsection 3.2) in some of the selected frequency components and M2 is obtained. The accrued noise due to embedding is minimized (Subsection 3.3) to obtain M3. The technique of IDHWT is applied on M3 to obtain stego values as shown in M4. But it is seen that some of the stego values contradict the image property. An adjustment technique (Subsection 3.4) is implemented to maintain the image property for all the stego values and the final set is M5.

In the extraction process, the embedding bit sequence is extracted from M6 which is obtained by applying DHWT on M5.

The algorithm for embedding is explained in Subsection 3.1. In order to enhance the effectiveness of hiding mechanism, a mathematical function is defined in Subsection 3.2 to generate the pseudo-random embedding positions. Subsection 3.3 defines an initiative taken to achieve high PSNR values in spite of embedding at dynamically variable LSB positions (0-3). Protective measures as explained in Subsection 3.4 are also taken to maintain proper image property of the stego image.

In case of extraction (Subsection 3.5), the embedded bit sequence is extracted by using the mathematical expression $w(k) = (s'(m,n) - s(m,n))/(\alpha \times s(m,n))$ to reconstruct the payload size, message digest and the payload data.

3.1 The Insertion Algorithm

This procedure for embedding is discussed in Algorithm 1.

Algorithm 1 Embedding Technique

Input: An image as cover and a payload (Image/Text/Audio/Encrypted Data).
Output: A stego image.
Steps:

1: A 160 bit (SHA-1) message digest is generated from the payload.
2: Steps 2.1 to 2.7 are repeated to fabricate the payload size (in bytes), generated message digest and the payload entirely into the cover image,

2.1 From the cover image, non-overlapping 4x4 blocks of pixels are read in row major sequence.

2.2 The transform technique is applied on the spatial blocks sequentially to obtain the corresponding frequency blocks.

2.3 Only the integer part of the target frequency coefficients of a block are chosen for embedding.

2.4 The payload bits are scanned one at a time and embedded in pseudo-random positions in each of the selected frequency components.

2.5 An adjustment technique is applied on some of the modified frequency components to reduce the degree of noise due to embedding.

2.6 The stego pixels are obtained by applying Inverse Discrete Wavelet Transform on the selected block.

2.7 Necessary readjustments are applied on an adverse stego block to maintain proper image property.

2.8 The correct spatial block is written back into the output image in the same location.

3: **End**.

3.2 Generation of Pseudo-random Location

This procedure for generating pseudo-random embedding positions is discussed in Algorithm 2.

3.3 Reduction of Embedded Noise

This procedure to reduce noise due to the embedding procedure is discussed in Algorithm 3.

3.4 The Extraction Algorithm

This procedure for extraction is discussed in Algorithm 4.

3.5 Procedure for Retention of Image Property

This procedure to preserve the image property after the embedding technique is discussed in Algorithm 5.

Algorithm 2 Pseudo-Random Position

As input we consider three parameters p, q and r respectively, where p is a 8 bit integer value representing the mean of the current block, q (i.e. 0/1) signifies the last embedded bit of the payload and r (i.e. 0-7) represents the position of q in the current payload data byte. The generated pseudo-random value is a 2 bit integer value.
Steps:

1: Starting from LSB, three bits of r are taken.
2: A sequence qr2r1r0 (e.g. 1011) is formed to generate pos1 = qr2 XOR r1r0.
3: pos1 is regenerated as pos1 = (p1p0 XOR p3p2) XOR ($pos1_1$ $pos1_0$).
4: Let, Q be a buffer to hold maximum N values and prevent successive repetitions of the same random values of pos1.
5: Execute the following statements to get the final value of pos1.
IF (there are empty spaces in Q) then
The present value of pos1 is inserted into Q.
ELSE
The present pos1 is compared with its existing values in Q.
IF (there is a difference at any position) then
The present value of pos1 is treated as the final value.
ELSE
The current value of pos1 is modified as,
IF (q = 0) then
pos1 = (complement of $pos1_0$) $pos1_1$
ELSE
pos1 = $pos1_0$ (complement of $pos1_1$)
This value of pos1 in inserted into Q and returned as the final position.
6: **End**.

Algorithm 3 Reduction of noise

The magnitude of the stego component may differ or remain same as the spatial value. If s = e (b, t) then the magnitude of the adjustment factor n is defined as n = |s - b|. The adjustment on s is done based on:
Steps:

1: If n = 0, no adjustment is done.
2: if n > 0, then adjustment is done with respect to the embedding position of the payload bit. The bits of the binary representation of s are altered (0 to 1 / 1 to 0) on the right (i.e. towards LSB) and/or left (i.e. towards MSB).
Note: s is the stego frequency component, e () is the embedding function, b is the bit to be embedded and t is the position of embedding.

Algorithm 4 Extraction Technique

Input: A stego image.
Output: The extracted payload (Image/ Text/ Audio/ Encrypted Data).
Steps:

1: The fabricated bits are to be extracted to reform the required information.
2: Steps 2.1 to 2.5 are repeated to reform the payload.
2.1 From the stego image, non- overlapping 4x4 blocks of pixels are read in row major sequence.
2.2 Two Dimensional Discrete Haar wavelet Transform is applied on the spatial blocks at a time to obtain the frequency blocks.
2.3 The integer part of the target frequency values is chosen and the pseudo-random positions are generated for extraction.
2.4 The embedded bits are extracted from the selected areas.
2.5 The extracted bits are properly arranged to form the payload size, message digest and the payload itself.
3: The 160 bit message digest (SHA-1) is generated from the extracted payload.
4: The generated and extracted message digests are compared to check the authenticity of the received payload.
5: **End**.

Algorithm 5 Preservation of image property

In a stego block, some pixel values may contradict the image property. For rectification, specific adjustments are made on the spatial pixel values and the embedding algorithm is repeated again on that block. By experimentation on a number of gray scale images a threshold value T has been derived for the proposed algorithm to control the adjustment procedure. The possible adjustments are:

1: When all the spatial values in a block are in the range [0, T] only and a negative stego pixel value generates, then the highest difference, i.e. [0- (highest -ve pixel value)] is added to all the spatial values.
2: When all the spatial values in a block are in the range [(255-T), 255] only and there is a generation of a stego pixel value above 255, then the highest difference [(highest + ve pixel value) - 255] is subtracted from all the spatial values.
3: When all the spatial values in a block are in the range [(0, T), ((255-T), 255)] and there is a generation of both negative stego pixel value and stego pixel value above 255. The two types of differences are calculated as 1) A = [0- (highest -ve pixel value)] and 2) B = [(highest + ve pixel value) - 255]. The actual difference is selected as diff = max (A, B). The value (2 x diff) is subtracted from all the spatial values in the range [((255-T), 255)] and diff is added to all the spatial values in the range [0, T].
4: When all the spatial values in a block are in the range [(0, T), ((255-T), 255)] and there is a generation of both negative stego pixel value and stego pixel value above 255. The differences are calculated as in case 3.The actual difference is selected as diff = A or B. The value (2 x diff) is subtracted from all the spatial values in the range [((255-T), 255)] and diff is added to all the spatial values in the range [0, T].
5: When all the spatial values in a block are in the range [0, 255] and there is a generation of both negative stego pixel value and stego pixel value above 255. The differences are calculated as in case 3.The actual difference is selected as diff = A or B. The value (2 x diff) is subtracted from all the spatial values in the range [((255-T), 255)], diff is added to all the spatial values in the range [0, T] and the mid-range, i.e. [(T+1), ((255-T) - 1)] values remain unchanged.

4 Experiment and Results

The algorithm has been experimented on a number of gray scale images in a system with the following hardware configuration: 4 GB of main memory, processor of at least 1 GHz clock speed, 1 GB of graphics memory and 4 GB of free disk space. Various fidelity tests are performed on a number of gray scale images to analyse the robustness of embedding, i.e. to check whether there is any perceptual distortion in the cover image after the embedding of secret data. The imperceptibility is measured in terms of Mean Squared Error (MSE) [28], Peak Signal to Noise Ratio (PSNR) [28], Image Fidelity (IF) and Structural Similarity Index Metric (SSIM) [37]. The quantifiers are defined as follows:

1) Mean Square Error (MSE): The average energy of the error difference between the test and the reference signal is computed using Equation (3)

$$MSE = \frac{1}{MN}\sum_{y=0}^{M-1}\sum_{x=0}^{N-1}[I(x,y) - I'(x,y)]^2 \quad (3)$$

Here I(x, y) and I'(x, y) are the pixel values in the cover and the stego image and M, N are the horizontal and vertical pixel dimensions of the cover image.

2) Peak Signal to Noise Ratio (PSNR):The ratio of the maximum intensity of a signal against the intensity of the corrupting noise that affects the fidelity aspect is calculated by using Equation (4)

$$PSNR = 20log_{10}\frac{255}{\sqrt{MSE}} \quad (4)$$

where the constant value 255 signifies the maximum intensity of a pixel having a colour depth of 8 bits.

3) Image Fidelity (IF): The measurement of the degradation level of a perceived image w.r.t a perfect image is done with the help of Equation (5)

$$IF = 1 - \frac{\sum_{y=0}^{M-1}\sum_{x=0}^{N-1}[I(x,y) - I'(x,y)]^2}{\sum_{y=0}^{M-1}\sum_{x=0}^{N-1}[I(x,y)]^2} \quad (5)$$

4) Structural Similarity Index Metric (SSIM): This procedure is sensitive to distortions that disintegrate the natural spatial correlation of an image and considered as the best possible method to evaluate image quality. It is evaluated as a product of luminance comparison function l (f, g), contrast comparison function c(f, g) and structural comparison function s(f, g) as shown in Equation (6).

$$SSIM(f,g) = l(f,g).c(f,g).s(f,g) \quad (6)$$

where,

$$\begin{cases} l(f,g) = \dfrac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \\ c(f,g) = \dfrac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \\ s(f,g) = \dfrac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \end{cases} \quad (7)$$

The first term, i.e. luminance comparison function used in Equation (7) measures the closeness of mean luminance (μ_f and μ_g) of the cover and stego images. The second term, i.e. contrast comparison function

measures the closeness of the contrast (measured by the standard deviation σ_f and σ_g) of the two images. The third term i.e. the structure comparison function measures the correlation coefficient between the two images f and g. Note that σ_{fg} is the covariance between f and g. The positive values of the SSIM index fall between [0,1] where 0 signifies no correlation between images and 1 means $f = g$. The positive constants C_1, C_2 and C_3 are used to avoid a null denominator.

The comparison between the three areas of embedding in terms of PSNR (in dB), IF and SSIM are shown in Table 1, Table 2 and Table 3 respectively. In case of area 1 and area 2 the size of the payload is 110x110 and in case of area 3 the size of the payload is 120x120. The size of each of the cover images is 512x512 for all the cases.

Table 1: Analysis of PSNR

Cover Images	Payload	PSNR(in dB)		
		Area I	Area II	Area III
Monalisa	Earth	37.19	37.16	37.22
Lenna		36.17	36.14	36.29
Baboon		37.40	37.42	37.49
Oakland		37.09	37.16	37.25
Woodlad		36.77	36.86	37.10
Peppers		36.10	36.12	36.40
Tiffany		34.15	34.10	34.63
Airplane		35.10	34.89	35.16
Sailboat		35.39	35.46	36.55
Average		**36.15**	**36.14**	**36.45**

Table 2: Analysis of IF

Cover Images	Payload	IF		
		Area I	Area II	Area III
Monalisa	Earth	0.9954	0.9958	0.9959
Lenna		0.9340	0.9334	0.9356
Baboon		0.9933	0.9938	0.9938
Oakland		0.9939	0.9947	0.9950
Woodlad		0.9984	0.9982	0.9983
Peppers		0.9859	0.9853	0.9864
Tiffany		0.9918	0.9921	0.9920
Airplane		0.9835	0.9836	0.9843
Sailboat		0.9959	0.9963	0.9964
Average		**0.9857**	**0.9859**	**0.9864**

It can be considered that the degradation level of a stego image is quite acceptable if the PSNR value is greater than 35 dB, i.e. the payload is almost invisible to HVS. Table 1 suggests that the proposed algorithm is quite successful in achieving a decent average PSNR value, i.e. 36.24 dB in spite of hiding quite a large volume of secret data simultaneously in both the levels of resolution. Moreover from Table 2 and Table 3 the average values of IF and SSIM, i.e. 0.9860 and 0.9948 respectively, proves that the stego image more or less resembles the cover image. Also, visual analysis of the testing images shows imperceptible distinction between the cover and the stego images as shown for the three areas of embedding in Figure 5, Figure 6 and Figure 7 respectively.

Table 3: Analysis of SSIM

Cover Images	Payload	SSIM		
		Area I	Area II	Area III
Monalisa	Earth	0.9990	0.9988	0.9988
Lenna		0.9963	0.9961	0.9959
Baboon		0.9959	0.9962	0.9964
Oakland		0.9913	0.9915	0.9915
Woodlad		0.9954	0.9960	0.9960
Peppers		0.9973	0.9972	0.9971
Tiffany		0.9851	0.9850	0.9844
Airplane		0.9949	0.9948	0.9948
Sailboat		0.9977	0.9985	0.9985
Average		**0.9947**	**0.9949**	**0.9948**

In addition to this, our proposed embedding algorithm can be implemented in all the middle frequency bands, irrespective of whether they are present in a combination of 1-level and 2-level of resolution or only in 2-level of resolution and the average value of PSNR suggest robust embedding. Moreover, performance comparison is also done with other similar existing algorithms as shown in Table 4.

Table 4: Comparative analysis of PSNR

No.	**Algorithm**	**Level of Resolution**	**Avg. PSNR (in dB)**
A	[3]	1	34.60
B	[27]	1	38.52
C	[2]	2	39.62
D	[12]	2	26.30
E	[6]	2	33.08
F	[23]	2	36.04
G	Proposed	2	36.44
H	Proposed	2	43.22
I	Proposed	2	45.73

From the above table it shows that as compared to 1-level resolution in A, with similar embedding capacity our proposed algorithm even with 2-level of resolution in G generates an enhanced PSNR value of 36.44 dB. Similarly, considering the same embedding capacity as in B, our algorithm with 2-level of resolution in I shows an increased PSNR value of 45.73 dB as compared to 38.52 dB

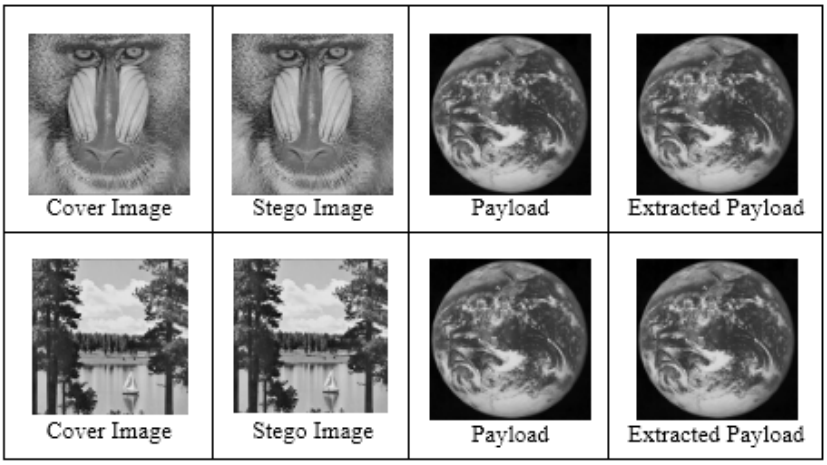


Figure 5: Embedding in Area I

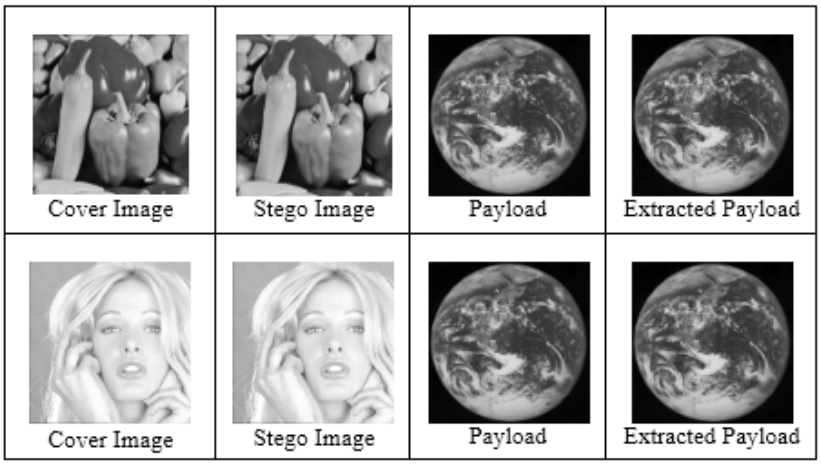


Figure 6: Embedding in Area II

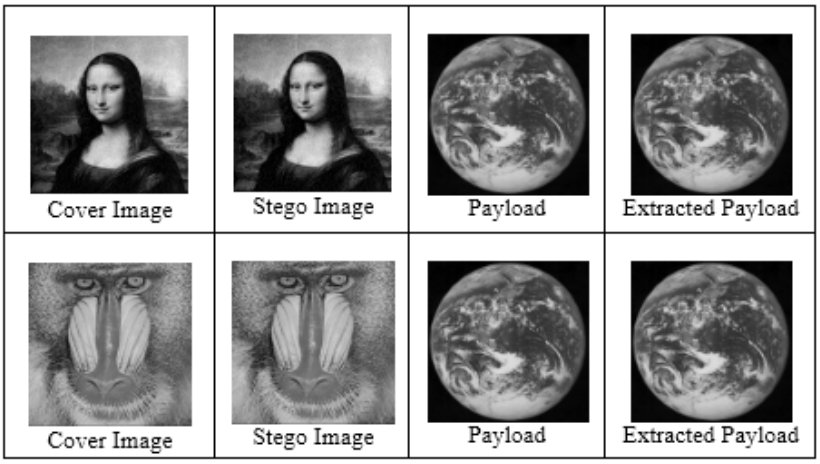


Figure 7: Embedding in Area III

Figure 8: Visual analysis of cover and stego images

in B even with 1-level of resolution. In addition to this, considering 2-level of resolution and similar embedding capacity our algorithm in H shows an escalation in PSNR value, i.e. 43.22 dB when compared with some similar algorithms like C, D, E and F having PSNR values of 39.62 dB, 26.30 dB, 33.08 dB and 36.04 dB respectively.

5 Resistance to Certain Attacks

5.1 Visual Attack

Initiative is imparted to resist visual attack by eliminating certain issues related to secret message like 1) embedding in a sequential order and 2) length is less than the maximum size of the bit plane. Figure 8 shows the visual interpretation of the magnified version of a source and the stego image of the three areas and it may be inferred that the algorithm can resist visual attack considerably.

5.2 Statistical Attack

This is similar to visual attack. The assumption is that the least significant bit of a cover image is random and may be replaced by a secret message is not necessarily correct. The basic concept is to compare between the frequency distribution and the theoretically expected distribution of a potential cover image. If the statistical profile of the new data does not with the standard data, then it probably contains a hidden message. In Figure 9 there is a detailed graphical comparison between the cover and stego signals with respect to the pixel intensity value vs. pixel number. It shows that the graph for stego image mostly overlaps the graph for cover image. Due to efficient adjustment of noise in the stego image, no noticeable changes occur after the concealment of the payload in the carrier image. So we may infer that the proposed algorithm is significantly robust in the domain of data authentication and different areas of comparison as shown in sub figures strongly establish the fact. Moreover as the cover and stego images are more or less identical and statistically similar, an unintended user will have confusion in implementing statistical attack.

5.3 Copy Attack

The objective is to copy a payload from one carrier signal to another. It is performed in two steps: 1) An estimation of the embedded payload is made from the stego image and 2) The estimated payload is copied from the stego image to a carrier signal to obtain a separate target stego image. Our proposed algorithm uses a self-defined mathematical pseudorandom function to establish a link between the payload and the cover image. As the function is only known to the authorized sender and receiver, the link can only be verified by an intended receiver during the extraction of the payload. In addition to this, the pseudorandom function also helps to make the payload a function of the original cover image which may cause a problem in terms of the marked target cover image. The elimination of copy attack also helps to resist protocol attack.

6 Conclusions

Social Implication: Personal identification and authentication demand the necessity of security, protection and access restriction which may be achieved using a biometric authentication system. The characteristic of biometric system is unique and cannot be lost or forgotten and the components used by biometric systems include fingerprints, hand geometry, iris, retina, facer, hand, vein etc.

To access sensitive and restricted areas in an office, we like to propose the facility of smart card for an individual's identity. A chip will be embossed on the identity card and the information content in the chip will be original template, entire biometric sensor, microprocessor and memory. This chip will function as per the operations of System-on-Card (SOC).

The card is prepared using two images namely: 1) Digital photo as cover image and 2) Image of his or hers retina as payload. These two images are fed as input to the proposed embedding algorithm to produce a stego image. The stego image is written into the memory of the chip and the identity card is prepared to be delivered to the individual. At the entry

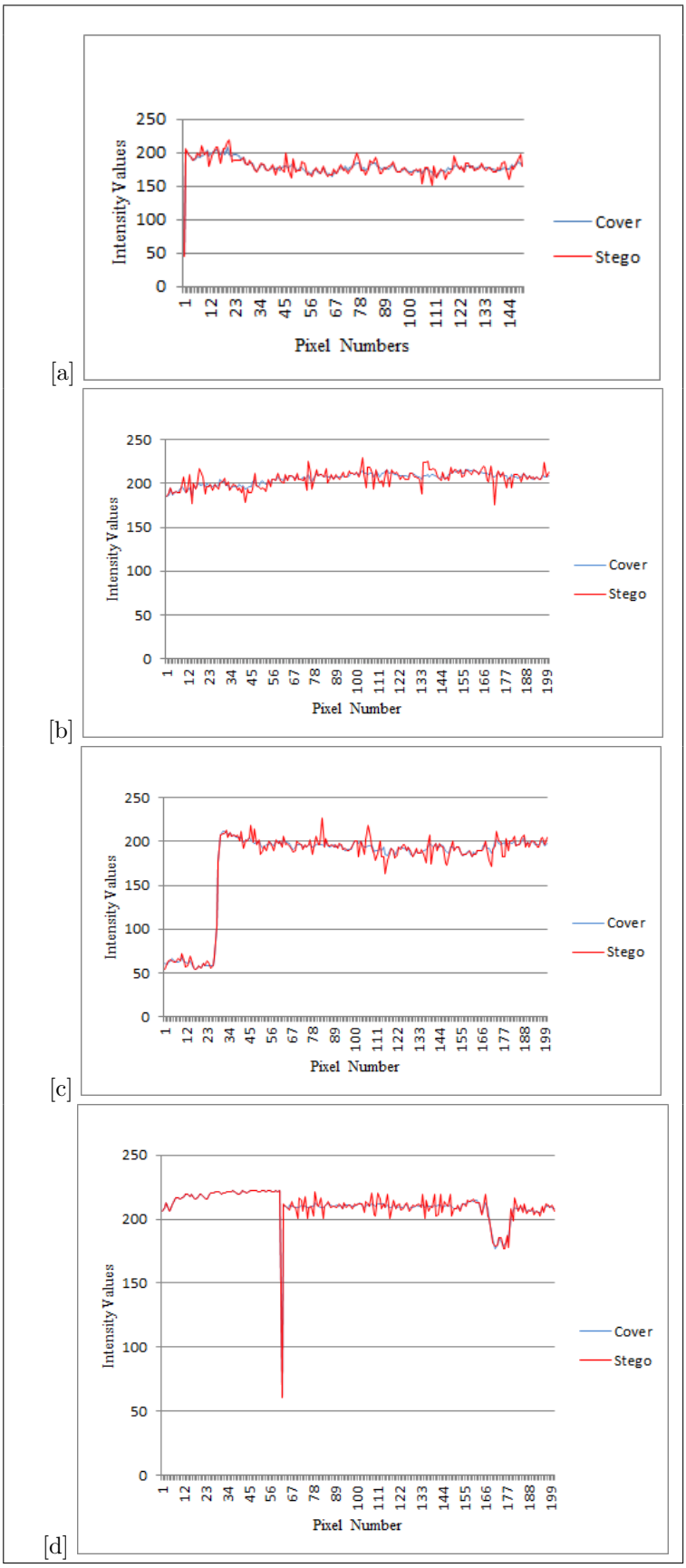


Figure 9: Statistical analysis of source and stego image pixels

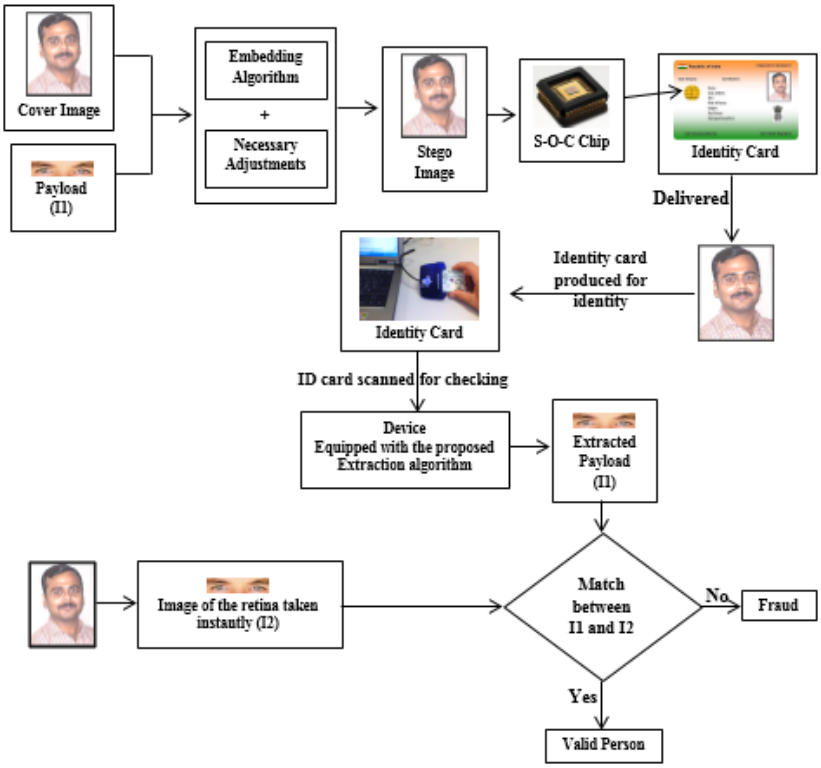


Figure 10: Authentication in smart card

point where the card is required to prove an individual's identity, the image of the retina of the individual is captured instantly. The embedded stego image is extracted by scanning the chip on the identity card. The payload i.e. original image of retina is extracted by using the extraction algorithm. The captured and the extracted images of the retina are matched to validate an individual's identity. It is pictorially represented in Figure 10 as.

Conclusions: This paper proposes a secured data authentication algorithm for the protection of copyright information. Embedding capacity, level of decomposition and the areas chosen for embedding are quite better than many existing methods. The PSNR values computed for the three proposed areas show effective results and the average values of IF and SSIM ensure the similarity between the cover image and the stego image. The extraction of the original payload bits is also difficult as they are not embedded directly into a fixed LSB position in the spatial domain of a cover image. In addition to this, the payload can be extracted [25] without the availability of the original cover image and the algorithm proves effective against some related attacks.

Future Studies and Recommendations: The proposed algorithm can be implemented using other wavelet frequency methods. Moreover color cover image can also be taken into consideration with embedding in other frequency bands.

Acknowledgments

The authors express their deep sense of gratitude to the faculty members of the Dept. of Engineering and Technological Studies, University of Kalyani, West Bengal, India, where the work has been carried out.The work has been financially supported by DST, PURSE.

References

[1] N. A. Abu, P. W. Adi, and O. Mohd, "Robust digital image steganography within coefficient difference on integer Haar wavelet transform," *International Journal of Video and Image Processing and Network Security*, vol. 14, no. 2, pp. 1–8, 2014.

[2] Anuradha and R. P. Singh, "Dwt based watermarking algorithm using Haar wavelet," *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 1, pp. 1–6, 2012.

[3] S. Bhattacharyya and G. Sanyal, "Data hiding in images in discrete wavelet domain using pmm," *Academic Journal, World Academy of Science, Engineering and Technology*, vol. 44, pp. 376–384, 2010.

[4] M. Borda and I. Nafornita, "Digital watermarking - principles and applications," in *Proceedings of*

The International Conference on Communications, pp. 41–54, 2004.

[5] P. A. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, pp. 532–540, 1983.

[6] A. Chawla and P. Shukla, "A modified secure digital image steganography based on DWT using matrix rotation method," *International Journal of Computer Science and Communication Engineering*, vol. 2, no. 2, pp. 20–25, 2013.

[7] P. Y. Chen and H. J. Lin, "A DWT based approach for image steganography," *International Journal of Applied Science and Engineering*, vol. 4, no. 3, pp. 275–290, 2006.

[8] Q. Cheng and T. S. Huang, "Robust optimum detection of transform domain multiplicative watermarks," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 906–923, 2003.

[9] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Transactions on Image Processing*, vol. 7, pp. 213–245, 1998.

[10] G. M. Davis and A. Nosratinia, "Wavelet-based image coding: An overview," in *Applied and Computational Control, Signals, and Circuits*, pp. 369–434, 1999.

[11] S. Dey, A. Abraham, and S. Sanyal, "An LSB data hiding technique using prime numbers," in *Third IEEE International Symposium on Information Assurance and Security*, pp. 101–106, 2007.

[12] Y. Dinesh and A. P. Ramesh, "Efficient capacity image steganography by using wavelets," *International Journal of Engineering Research and Applications*, vol. 1, no. 1, pp. 251–259, 2012.

[13] D. J. Field, "Wavelets, vision and the statistics of natural scenes," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 357, pp. 2527–2542, 1999.

[14] D. J. Field and N. Brady, "Wavelets, blur and the sources of variability in the amplitude spectra of natural scenes," *Vision Research*, vol. 37, pp. 3367–3383, 1997.

[15] V. Fotopoulos and A. N. Skodras, "A novel approach on transform domain watermarking against geometrical deformations," in *Proceedings of IEEE workshop on Signal Processing Systems Design and Implementation*, pp. 403–406, Nov. 2005.

[16] N. Ghoshal, S. Chowdhury, and J. K. Mandal., "A steganographic scheme for colour image authentication using z-transform (SSCIAZ)," in *Proceedings of The International Conference on Information System Design and Intelligent Applications*, pp. 209–216, Jan. 2012.

[17] N. Ghoshal and J. K. Mandal, "Controlled data hiding technique for color image authentication in frequency domain," in *Proceedings of The Second International Conference on Emerging Applications of Information Technology (EAIT'11)*, pp. 284–287, Feb. 2011.

[18] N. Ghoshal and J. K. Mandal, "Discrete fourier transform based multimedia color image authentication for wireless communication (dftmciawc)," in *Proceedings of The Second International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology*, pp. 1–5, 2011.

[19] N. Ghoshal, J. K. Mandal, and A. Khamrui, "A framework for block based image authentication," in *Proceedings of The Fourth IEEE International Conference on Industrial and Information Systems (ICIIS'09)*, pp. 343–348, Dec. 2009.

[20] N. Ghoshal, J. K. Mandal, A. Sarkar, and D. Chakrabarty, "Image authentication by hiding large volume of data and secure message transmission technique using mask," in *Proceedings of The IEEE International Advanced Computing Conference (IACC'09)*, pp. 3177–3182, Mar. 2009.

[21] M. Juneja and P. S. Sandhu, "Improved LSB based steganography techniques for color images in spatial domain," *International Journal of Network Security*, vol. 16, no. 6, pp. 452–462, 2014.

[22] S. A. Kasmani and A. R. Naghsh-Nilchi, "A new robust digital image watermarking technique based on joint DWT DCT transformation," in *Proceedings of The Third IEEE International Conference on Convergence and Hybrid Information Technology (ICCIT'08)*, pp. 539–544, Busan, Korea, 2008.

[23] L. Li, H. H. Xu, C. C. Chang, and Y. Y. Ma, "A novel image watermarking in redistributed invariant wavelet domain," *The Journal of Systems and Software*, vol. 84, no. 6, pp. 923–929, 2011.

[24] G. Manikandan, M. Kamarasan, and N. Sairam, "A new approach for secure data transfer based on wavelet transform," *International Journal of Network Security*, vol. 15, no. 2, pp. 106–112, 2013.

[25] Q. Mao, C. C. Chang, and T. F. Chung, "A reversible steganography suitable for embedding small amounts of data," *International Journal of Network Security*, vol. 16, no. 4, pp. 295–303, 2014.

[26] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR'88)*, pp. 19–43, 1988.

[27] P. Patil and D. S. Bormanel, "DWT based invisible watermarking technique for digital images," *International Journal of Engineering and Advanced Technology*, vol. 2, no. 4, pp. 603–605, 2013.

[28] S. Poobal and G. Ravindran, "The performance of fractal image compression on different imaging modalities using objective quality measures," *International Journal of Engineering Science and Technology*, vol. 2, no. 1, pp. 239–246, 2011.

[29] P. Raviraj and M. Y. Sanavullah, "The modified 2d-Haar wavelet transformation in image compression," *Middle-East Journal of Scientific Research*, vol. 2, no. 2, pp. 73–78, 2007.

[30] A. Said and W. A. Pearman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1303–1310, 1996.

[31] S. S. Sujatha and M. Mohamed Sathik, "A novel DWT based blind watermarking for image authentication," *International Journal of Network Security*, vol. 14, no. 4, pp. 223–228, 2012.

[32] S. S. Tamboli and V. R. Udupi, "Image compression using Haar wavelet transform," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3166–3170, 2013.

[33] Y. Y. Tsai, J. T. Chen, and C. S. Chan, "Exploring LSB substitution and pixel-value differencing for block-based adaptive data hiding," *International Journal of Network Security*, vol. 16, no. 5, pp. 363–368, 2014.

[34] T. Vanitha, A. D. Souza, B. Rashmi, and S. Dsouza, "A review on steganography - least significant bit algorithm and discrete wavelet transform algorithm," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 5, pp. 89–95, 2014.

[35] A. Verma, R. Nolkha, A. Singh, and G. Jaiswal, "Implementation of image steganography using 2-level DWT technique," *International Journal of Computer Science and Business Informatics*, vol. 1, no. 1, pp. 1–14, 2013.

[36] S. M. C. Vigila and K. Muneeswaran, "Hiding of confidential data in spatial domain images using image interpolation," *International Journal of Network Security*, vol. 17, no. 6, pp. 722–727, 2015.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[38] A. B. Watson, "Efficiency of a model human image code," *Journal of the Optical Society of America*, vol. 4, no. 12, pp. 2401–2417, 1987.

[39] C. Yang, C. Weng, S. Wang, and H. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 488–497, 2008.

Anirban Goswami is currently working as Asstt. Professor in Techno India (An Engineering College under West Bengal University of Technology), Kolkata, West Bengal, India and pursuing his research work in the Department of Engineering and Technological Studies, University of Kalyani, West Bengal, India. He has 15 years of teaching experience and had contributed in more than 10 graduate level projects. He has 7 international conference and 2 international journal publications.

Dr. Nabin Ghoshal is a Professor in the Department of Engineering and Technological Studies, University of Kalyani, West Bengal, India. He received a Bachelor degree in Mathematics from the University of Calcutta in 1994. He got Master degree in Computer Applications from the University of North Bengal in 1998 and also obtained M. Tech. degree in Computer Science and Engineering from the University of Kalyani in 2005 respectively. He received his Ph. D. degree in Computer Science and Engineering from the University of Kalyani in 2011. Dr.Ghoshal is committed to professional teaching and research work for many years, accumulated rich experience in teaching and research on topics namely Steganography, Watermarking, Visual Cryptography, Security, Visual Cryptography through Steganography, Copyright protection and Legal Document Authentication (Audio/ Video). He has 41 research papers in various international journals and national and international conferences. He wrote a book entitled "Steganographic Techniques and Application in Document Authentication".

Sequential Secret Sharing Scheme Based on Level Ordered Access Structure

Dileep Kumar Pattipati[1]*, Appala Naidu Tentu[2], V. Ch. Venkaiah[3], Allam Appa Rao[2]
(Corresponding author: Appala Naidu Tentu)
Computer Science and Engineering, IIT Madras, Chennai-600036, India[1]
CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science, University of Hyderabad Campus[2]
Hyderabad-500046, India
School of Computer and Information Sciences, University of Hyderabad, Hyderabad-500046, India[3]
(Email: naidunit@gmail.com)
(Received Dec. 26, 2014; revised and accepted Jan. 16 & Mar. 4, 2015)

Abstract

In Software Industry an application can be released to production only after it has gone through Unit testing, followed by Integration testing, then System testing and finally Acceptance testing. Note here that without the completion of unit testing, integration testing cannot be started and similarly without the completion of integration testing, system testing cannot be started and so on. That is the ordering is important. To realize this or similar kind of activity we need a hierarchial access structure that has in built ordering among the levels. Existing access structures fail to realize this scenario as they are short of enforcing the required ordering. The purpose of this paper is to propose an access structure that caters to this kind of scenarios and come up with schemes that realize this access structure. We call this new access structure as Level Ordered Access Structure(LOAS) and the schemes that realize this access structure as Level Ordered Secret Sharing(LOSS) schemes.

Keywords: Level ordered access structure, level ordered secret sharing, ordered hierarchial, threshold secret sharing

1 Introduction

Secret sharing is a cryptographic primitive which is used to distribute a secret among a group of players. It is simply a special form of key distribution [14]. The distribution is such that any group of authorized players can always reconstruct the secret, whereas an unauthorized group can never obtain any information about the secret. The first secret sharing scheme was designed independently by Shamir in [12] and Blakley in [4]. The approach in [12] relies on Lagrange polynomial interpolation, whereas the scheme in [4] is geometric and uses the concept of intersecting hyperplanes.

The Access Structure of a secret sharing scheme is the set of all groups which are allowed to reconstruct the secret. It is denoted by Γ. The elements of an access structure are referred to as the authorized sets and the rest are called unauthorized sets. The set of all unauthorized sets is called the Adversary structure. The adversary structure will be denoted by $\bar{\Gamma}$. An access structure is called monotone if it satisfies the following criteria.

1) $(A \in \Gamma) \wedge (A \subseteq B) \Longrightarrow B \in \Gamma$;

2) $(A \in \bar{\Gamma}) \wedge (B \subseteq A) \Longrightarrow B \in \bar{\Gamma}$.

We assume that Γ only contains the minimal allowed groups which can recover the secret. Similarly, $\bar{\Gamma}$ only contains maximal adversarial groups which cannot recover the secret.

Several access structures have been proposed in the literature. The primitive access structure is the (t, n)-threshold access structure. In a (t, n)-threshold access structure, there are n shareholders. An authorized group consists of any t or more participants and any group of at most $t-1$ participants is an unauthorized group.

Threshold schemes are suitable for situations in which each player is assigned the same trust. In most practical situations, the degree of trust assigned to a player can differ based on the authority of the player. Simmons [13] introduced *multilevel t_i-out-of-n_i* and *compartmented t_i-out-of-n_i* secret sharing schemes to model secret sharing in some practical situations wherein the trust is not distributed uniformly over the set of all players

In Multilevel secret sharing, a set of players is partitioned into disjoint levels. Players at lower levels have more importance than players at higher levels. Each level i contains n_i players. So, the levels form a hierarchial structure. Hence multilevel secret sharing is also called hierarchial secret sharing. There are two types of multilevel

*Work done when the author was at University of Hyderabad

access structures: disjunctive multi-level access structure introduced by Simmons [13] and conjunctive multi-level access structure by Tassa [16].

In disjunctive multi-level access structure *any* t_i players of the i^{th} level can recover the secret. When the number of cooperating participants from the i^{th} level is smaller than t_i, say r_i, then $t_i - r_i$ participants can be taken from lower levels.

In conjunctive multi-level access structure *every* group of t_i players on the i^{th} level must cooperate to recover the secret. When the number of cooperating participants from the i^{th} level is smaller than t_i, say r_i, then $t_i - r_i$ participants can be taken from lower levels. A related signcryption scheme [18] for hierarchial groups is studied in [1].

In Compartmented secret sharing, a set of players is partitioned into disjoint compartments. The secret is distributed such that reconstruction of the secret requires cooperation of at least t_i players from the i^{th} compartment. In this context, let us recall the example presented by Simmons in [13]. Let two countries agree to control the recovery of the secret (which may initiate a common action) by a secret sharing scheme. The secret can be recreated only if at least two participants from both compartments pool their shares together.

Generalized access structure is the far reaching generalization of the access structures discussed above. Let U be a set of n participants and 2^U be its power set. Generalized access structure refers to the case when the collection of authorized subsets of U may be any collection $\Gamma \subseteq 2^U$ having the monotonicity property.

A secret sharing scheme is a perfect realization of Γ [15] if for all authorized sets $A \in \Gamma$, the users in A can always reconstruct the secret, and for all unauthorized sets B not in Γ, the users in B collectively cannot obtain any information about the secret. Schemes that satisfy this criteria is commonly referred to as unconditionally secure schemes.

The information rate, ρ_i, for participant i is defined as the ratio of the length of the secret, expressed in bits, to the length of the share, also expressed in bits i.e.

$$\rho_i = \frac{\log_2 |\text{secret}|}{\log_2 |\text{share}|}.$$

The information rate ρ of the scheme is defined as $\rho = \min\{\rho_i : i \text{ is a participant of the scheme}\}$.

A well known fact in secret sharing is that the size of a share is at least the size of the secret. Therefore, the information rate of the participant and hence the information rate of the scheme are both bounded between 0 and 1. Schemes with maximum information rate are desirable [15]. Schemes with information rate 1 are called ideal schemes [15]. The relationship between permutations and ideal secret schemes is studied in [10].

1.1 Our Contribution

Many applications require that secrets be reconstructed in a well-defined order. For example, in banks, a cheque has to be cleared first by the clerk, then by the cashier and finally by the manager. The order has to be strictly enforced. These applications require ordering theory to be introduced into an access structure. It may appear that this problem can be solved by using Multistage secret sharing, but in fact it is not. Refer Section 2.2 for details and Example 1 for a concrete example. To the best of our knowledge, this is the first paper to bring ordering theory into access structures.

A formal definition of proposed Level ordered Access structure (LOAS) is presented in the paper. Also, an ideal secret sharing scheme that realizes this access structure is presented. The scheme is similar in spirit to the compartmented secret sharing scheme proposed by Brickell [5], but differs in the way the partial secrets are combined to recover the secret. The way we combine ensures ordering among the levels, which is the main objective behind Level ordered secret sharing.

1.2 Outline of the Paper

Formal Definition of level ordered access structure is presented in Section 2. The difference between Level ordered secret sharing schemes (LOSS) and other extensions of Shamir secret sharing especially Hierarchial secret sharing are discussed in Section 2. An interesting relationship between generalized access structures and LOAS is discussed in Appendix 4. LOAS and its properties are discussed in Section 3. Section 3 also discusses the modification of LOAS to include a virtual player, which in turn enables to prove the existence of an ideal scheme for the LOAS. In addition, an ideal scheme and the properties of the LOSS scheme especially homomorphic properties are presented in Section 3. Finally we conclude the paper with possible directions for future work in Section 4.

2 Formal Definition of LOAS

In LOAS, a set of players are partitioned into different levels and each level is associated with a threshold. Also there is an ordering defined on the levels. During reconstruction, if the players submit shares according to the specified order, then the actual secret should get reconstructed. Formally the proposed Level ordered Access structure is as follows.

Definition 2.1 *Let U be a set of n participants and let $U_1, U_2, \cdots U_m$ be a partition of the set U. Also let b_i be a boolean variable, which we call the activation index associated with the i^{th} level U_i, $1 \leq i \leq m$. Define S_i, recursively, to be an authorized set corresponding to the i^{th} level if*

1) *$S_i \subseteq U_i$ and $|S_i| \geq t_i$,*

2) *$\exists$ an authorized set (S_{i-1}) whose activation index (b_{i-1}) is True, where $b_0 = T$ and $S_0 = \emptyset$.*

I.e., there is an authorized set S_{i-1} of $(i-1)^{th}$ level and the truth value of the corresponding activation index b_{i-1} is true.

A authorized sets of LOAS are the authorized sets of level m.

2.1 Relationship Between LOAS and Hierarchical and Compartmented Access Structures

There are a number of related definitions of access structures like Hierarchial and Compartmented access structures. Following arguments (discussion) explains that these access structures are different from the LOAS defined above.

Definition 2.2 *Disjunctive hierarchical access structure is a multipartite access structure in which each level L_i is assigned with a threshold t_i, $1 \leq i \leq m$, and the secret can be reconstructed when, for some i, there are at least t_i shareholders who all belong to levels smaller than or equal to L_i. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap (\cup_{j=1}^{i} U_j)| \geq t_i, \text{ for some } i \in \{1, 2, \cdots, m\}\}.$$

Definition 2.3 *Conjunctive hierarchical access structure is a multipartite access structure in which each level L_i is assigned with a threshold t_i for $1 \leq i \leq m$, and the secret can be reconstructed when, for every i, there are at least t_i shareholders who all belong to levels smaller than or equal to L_i. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap (\cup_{j=1}^{i} U_j)| \geq t_i, \text{ for every } i \in \{1, 2, \cdots, m\}\}.$$

Note that in Hierarchical secret sharing, players can be taken from lower levels and this is not permissible in LOAS. Also LOAS defines a sequence of levels where lower levels have to submit their shares before higher levels, whereas such requirement is absent in hierarchical secret sharing.

Definition 2.4 *Compartmented access structure is a multipartite access structure in which each compartment is assigned with a threshold t_i, $1 \leq i \leq m$, and the secret can be reconstructed when, for every i, there are at least t_i shareholders from U_i and a total of at least t_0 participants from all the compartments. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap U_i| \geq t_i, \text{ for every } i \in \{1, 2, \cdots, m\} \text{ and } |V| \geq t_0\}.$$

where $t_0 \geq \sum_{i=1}^{m} t_i$. Compartmental secret sharing and LOAS bear a similarity. In fact, we'll see in Section 3 that the elementary access structure in Level ordered access structure is a Compartmented access structure. There is no concept of ordering among the compartments in a Compartmented access structure.

2.2 Relationship Between Multistage Secret Sharing and LOAS

In a Multistage secret sharing (MSS) scheme [8, 7, 20], shares are distributed to users so that k secrets can be reconstructed, one at each stage. Each participant receives a share known as master share. In each of the stages, a shadow share is computed for each user based on his master share. The shadow shares are used to reconstruct the secret at that stage. Also each stage uses some public values. Note that these methods reconstruct the secrets sequentially. Literature also offers methods that reconstruct all the secrets simultaneously [6, 20]. These methods are known as parallel secret reconstruction methods.

We would like to call the above traditional method of multistage secret sharing as "Loose sequential secret sharing" as a secret at level L_i may be computed without the knowledge of the secret at level L_{i-1}. Also this method supports parallel secret reconstruction.

The LOAS secret sharing scheme described in this paper can be called as "Strict sequential secret sharing" as the secret at level L_i requires the knowledge of secret at level L_{i-1} (See Section 2.3 for our idea of realizing LOAS). It is straightforward to infer that strict sequential secret sharing cannot support parallel secret reconstruction.

More formally, the distinction between loose and strict sequential secret sharing schemes(a scheme that involves a secret at each level) can be made as follows. Any sequential secret sharing scheme can be characterized by two parameters: the first parameter is a triple $(Id, \Gamma_{Id}, s_{Id})$, where Id is the stage or level identity, Γ_{Id} is the access structure for the stage and s_{Id} is the (partial) secret associated with the stage. The second parameter is a permutation on the stage Ids describing the valid order of secret reconstruction.

In traditional MSS Γ_{Id} is same for all stages and the permutation is often left unspecified to allow flexibility. The MSS schemes are flexible to allow the secret reconstruction of a random stage without reconstructing secrets in previous stages and also support parallel reconstruction. In LOAS schemes, there is no flexibility and the (partial) secrets need to be recovered in the specified order.

A simple modification to an existing MSS scheme, like addition of previous stage secret to current stage (t, n) Shamir secret, cannot accomplish the requirements of LOAS, as can be seen from Example 1.

There exists an interesting relationship between Generalized Access Structures and LOAS based on discrete mathematics concept, POSET. But, in order to continue the flow, we defer the discussion to Appendix 4.

The conclusion is that LOAS is different from Hierarchial secret sharing, Compartmental secret sharing, and Multistage secret sharing. LOAS is a recursive set of access structures.

2.3 Realization of LOAS: An Overview

This section proposes an overview on the realization of LOAS. Specific implementation of the scheme is given in Section 3.

In our implementation, a partial secret s_i is associated with each level L_i. The partial secret in the last level is the actual secret of the scheme i.e $s_m = s$. The players at level L_i are allowed to reconstruct the partial secret s_i only after the players at level L_{i-1} have reconstructed the partial secret s_{i-1}.

3 Realization of Level Ordered Access Structure

In this section, the properties of LOAS are examined and a scheme which realizes the Level ordered access structure is given.

3.1 Virtual Player

A way of realizing the level ordered access structure is by adding a virtual player at each level except the first level. The partial secret at each level acts as share of the virtual player in the next level. The virtual player along with the threshold access structure of that level forms the modified access structure at that level. The addition of virtual player ensures that the secrets are reconstructed in specified order.

We define an elementary access structure for a level L_i to be the conjunction of a virtual player(P_i') and a (t, n) threshold access structure. For example, if a level L_i is associated with a (2,3) threshold access structure for players $P = \{P_1, P_2, P_3\}$ and the virtual player of the level is P' then the modified elementary access structure is

$$\begin{aligned}\Gamma &= P'(P_1P_2 + P_1P_3 + P_2P_3)\\ &= P'P_1P_2 + P'P_1P_3 + P'P_2P_3.\end{aligned}$$

One of the widely studied properties of the access structures is whether an ideal scheme exists for a given access structure or not. The following Theorem 2, establishes that the elementary access structure is an ideal access structure. Proof of this theorem is based on the the following theorem, which talks about the existence of an ideal scheme of an access structure, is due to Stinson [15].

Theorem 1 *If the vector corresponding to the dealer can be expressed as a linear combination of the vectors in every authorized set, then there exists an ideal scheme for the corresponding access structure.*

Theorem 2 *An ideal scheme exists for the elementary access structure.*

Proof: Let $GF(q)^d$ denotes the vector space of all d-tuples over $GF(q)$, where q is a prime power and $d \geq 2$. Define $d = t + 1$, where t is the threshold of the (t, n) threshold access structure. Let

$$\phi(P_i) = (0, 1, x_i, x_i^2, \cdots, x_i^{t-1})$$

for $1 \leq i \leq n$, where x_i is the x-coordinate given to P_i. Also, let

$$\begin{aligned}\phi(D) &= (1, 1, 0, \cdots, 0)\\ \phi(P') &= (1, 0, 0, \cdots, 0).\end{aligned}$$

Without loss of generality, let $(P_{i_1}, P_{i_2}, \cdots, P_{i_t}, P')$ be an authorized set. Also let $a_1, \cdots, a_t, a'$ be the coefficients chosen from $GF(q)$. Hence,

$$\begin{aligned}\phi(D) &= a_1\phi(P_{i_1}) + a_2\phi(P_{i_2}) + \cdots + a_t\phi(P_{i_t}) \\ &\quad + a'\phi(P') \qquad (1)\\ (1, 1, 0, \cdots, 0) &= \sum_{j=1}^{t} a_j(0, 1, x_{i_j}, x_{i_j}^2, \cdots, x_{i_j}^{t-1}) \\ &\quad + a'(1, 0, 0, \cdots, 0). \qquad (2)\end{aligned}$$

It can be easily seen from that $a' = 1$. The remaining set of equations can be expressed in matrix form as follows:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_{i_1} & x_{i_2} & \cdots & x_{i_t} \\ x_{i_1}^2 & x_{i_2}^2 & \cdots & x_{i_t}^2 \\ \vdots & \vdots & & \vdots \\ x_{i_1}^{t-1} & x_{i_2}^{t-1} & \cdots & x_{i_t}^{t-1} \end{pmatrix} \times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_t \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

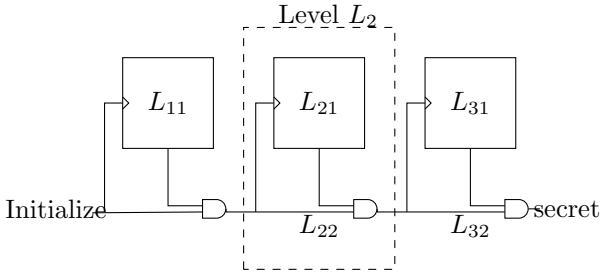
Since, the coefficient matrix is a Vandermonde matrix, its determinant is non-zero. So, the system has a unique solution. That is the vector $(0, 1, 0, \cdots, 0)$ can be expressed as a linear combination of the vectors of an authorized set.

3.2 Proposed Scheme

A look at virtual player concept reveals that each elementary access structure has two compartments. The first compartment is a (t, n) threshold access structure, and the second compartment has only a virtual player. We denote the j^{th} $(j = 1, 2)$ compartment of a level L_i with L_{ij}. So our scheme may be visualized as in the following block diagram.

3.3 Block Diagram

The LOAS can be shown in the form of a block diagram as shown below.



In the block diagram, the AND gate symbol is generic and it can be replaced with an XOR gate or an Adder (provides boolean addition of the two inputs) etc. In our algorithms below, we consider it to be an adder.

Let F_q be the ground field from which the shares and the secrets are chosen. Given the secret, the **Algorithm Share** assigns partial secrets to the levels of the access structure and subsequently to the players in the levels.

3.4 Algorithm Share

Let s be the secret, and i be the level index. Choose $s_1, \cdots, s_m$, so that $s_m = s$.

1) Initialize the partial secret of the last level s_m to s and level index i to m.

2) For each level $L_i (i > 1)$ with partial secret s_i do the following

   a. Assign s_{i-1} be the share of the virtual player at level i. Shares are assigned to the players in level L_{i1} based on Shamir's scheme [12] with $s_i - s_{i-1}$ as secret.

   b. Decrement the level index by 1 so that i becomes $i - 1$.

3) Assign shares to players in level L_1 based on Shamir's scheme [12] with s_1 as the secret.

3.5 Algorithm Reconstruct

1) The Shamir secret sharing scheme is used to generate partial secret, s_1 from the level L_1. The generated partial secret is the share of the virtual player in next level L_2. Initialize level index i to 2.

2) For each level L_i do the following

   a. The Shamir secret sharing scheme is used to generate secret from the first compartment L_{i1}, which is added with the share of the virtual player to generate the partial secret of level L_i. The generated partial secret is the share of the virtual player in the next level, i.e., the share of $L_{(i+1)2}$.

   b. Increment the level index i to become $i+1$, if $i < m$. Otherwise, return the partial secret of level L_m. This partial secret is the desired secret.

Remark: Note that there are two key operations in the proposed scheme. The first one is assigning the secret of stage $i (i < n)$ as the share of the virtual player in stage $i + 1$ and the other one is addition of partial secrets of stages i and $i + 1$ to provide the secret of stage $i + 1$. Both these operations are required to ensure ordering in the proposed scheme. Two examples are provided, each of which, tries to construct a scheme with only one of the above operations and fails to enforce the ordering.

Example 1 *(Considers only addition operation and excludes virtual player) Suppose that there are x stages and the order of secret reconstruction is $(s_1, \cdots, s_m)$ from left to right. The actual secret s is recovered only if the partial secrets are recovered in the specified order. Let $s_m = (t, n)$ Shamir $(s'_m) + s_{m-1}$, where s'_m is the partial secret recovered by stage m using Shamir secret sharing and $s_1 =$ Shamir (s'_1).*

From the definition of LOAS we have

$$\begin{aligned} s &= s_m \\ &= (t,n)Shamir(s'_m) + s_{m-1} \\ &= (t,n)Shamir(s'_m) + (t,n)Shamir(s'_{m-1}) + s_{m-2} \\ &= (t,n)Shamir(s'_m) + (t,n)Shamir(s'_{m-1}) + \cdots \\ &\qquad (t,n)Shamir(s'_1). \end{aligned}$$

Note that the final secret is simply the addition of the partial secrets of all the stages. So the actual secret can be constructed by any of the possible n! permutations with n stages. But according to the definition of LOAS, the secret should be recovered only if the partial secrets are reconstructed in the specified order.

Lemma 1 *The proposed scheme is perfect.*

Proof: It follows directly from the reconstruction algorithm that an authorized set can recover the secret. Any maximal unauthorized set B consists of $\sum_{i=1}^{m} t_i - 1$ players, where t_i is the threshold of the level L_i. So there exists a level L_j such that the number of corroborating players from that level fall below the threshold i.e., $B \cap L_j < t_j$. To find the partial secret of the level L_{j1}, we need t_j equations. But, the players from the level L_j provide a maximum of $t_j - 1$ shares. As the number of unknowns are less than the number of equations, there exists infinitely many solutions(i.e., $|F_q|$) for the secret value. Hence any maximal unauthorized set cannot obtain any information about the secret.

Theorem 3 *The proposed scheme is Level ordered. i.e., partial secrets are recovered in the specified order.*

Proof: We prove the theorem by the induction on levels. If there is only one level, the reconstruction algorithm returns the secret of the first level and terminates. Let the partial secret be recovered correctly for the k^{th} level (induction hypothesis). As per the construction, the first compartment of level $k+1$ implements Shamir secret sharing and provides the first input to the adder. As per the induction hypothesis, the second input is provided by the partial secret of the k^{th} level. Now the Adder can reconstruct the partial secret s_{k+1}. Hence, the partial secret in level L_{k+1} is reconstructed only after the partial secret in level L_k.

3.6 Properties of LOSS

3.6.1 Comparison with the Compartmental Access Structure

As can be seen from the virtual player concept that each elementary access structure other than the one at first level is a compartmental access structure with two compartments. The first compartment is a (t, n) threshold, the second compartment is a $(1, 1)$ threshold and the global threshold is $t + 1$. Note that sum of the individual thresholds is the global threshold. The elementary access structure in LOAS is a special case of the compartmental access structure in which sum of the individual thresholds is the global threshold.

3.6.2 LOSS is a Prepositioned Scheme

Prepositioned schemes [14] were introduced by Simmons and has two essential features:

Privacy. It should be possible to preposition all of the private information needed for the shared control subject to the condition that even if all of the participants were to violate the trust of their position and collaborate with each other, they would have no better chance of recovering the secret information than an outsider has of guessing it.

Activation. It should be possible to activate the shared control scheme once it is in place by communicating a single share of information, and for many applications, it should also be possible to reveal different secrets (using the same prepositioned private pieces of information) by communicating different activating shares of information.

LOSS is one of the best examples of prepositioned secret sharing schemes. The partial secret at level L_i is reconstructed only after the partial secret at level L_{i-1} is reconstructed. The partial secret at level L_{i-1} together with the activation index acts as activation information for the players at level L_i (Activation property). Without the partial secret at level L_{i-1}, the players at level L_i would have no better chance of recovering the secret information than an outsider has of guessing it (Privacy property).

3.6.3 Homomorphic Property of LOSS

The Homomorphic property of a secret sharing scheme allows to reconstruct the composition of secrets from the composition of corresponding shares without revealing anything about the individual secrets. Recovery of the partial secret at each level L_i in the reconstruction alogrithm of LOSS scheme comprises of two steps.

1) Shamir reconstruction algorithm to reconstruct the secret of the first compartment L_{i1};

2) Addition of secrets of levels L_{i-1} and L_{i2} to calculate the secret of the level L_i.

Shamir's scheme is homomorphic with respect to (+, +) [3] and the second operation is trivially homomorphic. Therefore, the proposed LOSS scheme is homomorphic with respect to (+, +).

4 Conclusion

This paper proposed an access structure that closely resembles the known access structures such as conjunctive hierarchial access structure and compartmental access structure. We call the proposed access structure as the Level Ordered Access Structure(LOAS). Unlike existing access structures; wherein there is no concept of ordering, LOAS enforces ordering and it is a sequence of threshold access structures.

It is easy to visualize applications of LOAS in variety of areas such as software testing, prepartion of cheques, drafts in banks etc. The paper presented a formal definition of LOAS and a model for realizing LOAS. The paper also analyzed the existence of an ideal scheme for the proposed LOAS and presented an ideal scheme for the same.

The side affects of cheating [17] by a player in the i^{th} level should be studied. Creating cheating models and analyzing the repercussions can be one direction for future work. To make the scheme secure against cheating, either a verification scheme [2] or a robust scheme [11] can be introduced. Designing such a scheme can be another direction for future work.

Acknowledgments

The authors gratefully acknowledge Prof.V.N.Sastry of IDRBT for his insights into the problem statement.

References

[1] A. Basu1, I. Sengupta1, and J. K. Sing, "Cryptosystem for secret sharing scheme with hierarchical groups," *International Journal of Network Security*, vol. 16, no. 6, pp. 455–464, 2013.

[2] M. Ben-Or, S. Goldwasser, A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC'88)*, pp. 1–10, New York, NY, USA, 1988.

[3] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret (extended abstract)," in *Advances in Cryptology (CRYPTO'86)*, LNCS 263, pp. 251–260, Springer, 1987.

[4] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the 1979 AFIPS National Computer Conference*, vol. 48, pp. 313–317, June 1979.

[5] E. F. Brickell, "Some ideal secret sharing schemes," in *Advances in Cryptology (EUROCRYPT'89)*, LNCS 434, pp. 468–475, Springer, 1990.

[6] H. Y. Chien and J. K. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics*, vol. 83, pp. 2762–2765, Sept. 2000.
[7] L. Harn, "Comment on "multistage secret sharing based on one-way function," *Electronics Letters*, vol. 31, pp. 262, Feb. 1995.
[8] J. He and E. Dawson, "Multistage secret sharing based on one-way function," *Electronics Letters*, vol. 30, pp. 1591–1592, Sept. 1994.
[9] K. M. Marin, *Discrete Structures in the Theory of Secret Sharing*, Ph.D. Thesis, University of London, 1991.
[10] 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.
[11] P. Rogaway and M. Bellare, "Robust computational secret sharing and a unified account of classical secret-sharing goals," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pp. 172–184, New York, NY, USA, 2007.
[12] A. Shamir, "How to share a secret," *Communications of ACM*, vol. 22, pp. 612–613, Nov. 1979.
[13] G. J. Simmons, "How to (really) share a secret," in *Advances in Cryptology (CRYPTO'88)*, LNCS 403, pp. 390–448, Springer, 1990.
[14] G. J. Simmons, "Prepositioned shared secret and/or shared control schemes," in *Advances in Cryptology (EUROCRYPT'89)*, LNCS 434, pp. 436–467, Springer, 1990.
[15] D. R. Stinson, "An explication of secret sharing schemes," *Designs, Codes and Cryptography*, vol. 2, no. 4, pp. 357–390, 1992.
[16] T. Tassa, "Hierarchical threshold secret sharing," *Journal of Cryptology*, vol. 20, pp. 237–264, Apr. 2007.
[17] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 2, pp. 133–138, 1988.
[18] M. Toorani and A. A. B. Shirazi, "Cryptanalysis of an ecient signcryption scheme with forward secrecy based on elliptic curve," *International Journal of Network Security*, vol. 10, no. 1, pp. 51–56, 2010.
[19] Wikipedia, *Partially Ordered Set*, The Free Encyclopedia, June 22, 2004. `http://http://en.wikipedia.org/wiki/Partially\_ordered\_set`
[20] T. Y. Yang, C. C. Chang and M. S. Hwang, "A (t,n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, pp. 483–490, 2004.

Appendix A.

Relationship Between Generalized Access Structures and LOAS

For mathematical background on POSET, Chains and Antichains the reader is referred to [19]. The following lemma, which is due to Martin[9] states an important relationship between the antichains of a POSET and generalized access structures.

Lemma 2 *Each of the antichain of a POSET P defines a generalized access strucutre.*

Example 2 *The antichains of the POSET* $P = \{\emptyset, \{x\}, \{y\}, \{z\}, \{x,y\}, \{y,z\}, \{z,x\}, \{x,y,z\}\}$ *are*
$\{\emptyset, \{\emptyset\}, \{\{x\}\}, \{\{y\}\}, \{\{z\}\},$
$\{\{x\},\{y\}\}, \{\{y\},\{z\}\}, \{\{z\},\{x\}\}, \{\{x\},\{y\},\{z\}\},$
$\{\{x,y\}\}, \{\{y,z\}\}, \{\{z,x\}\},$
$\{\{x\},\{y,z\}\}, \{\{y\},\{x,z\}\}, \{\{z\},\{x,y\}\},$
$\{\{x,y\},\{y,z\}\}, \{\{x,y\},\{z,x\}\}, \{\{y,z\},\{z,x\}\},$
$\{\{x,y\},\{y,z\},\{z,x\}\}, \{\{x,y,z\}\}\}$.

In the above example, excluding the empty set and the set containing empty set, the rest of the antichains define a generalized access structure. For example, the antichain $\{\{x\},\{y,z\}\}$ defines an access structure $\Gamma = x + yz$.

Define the operator RECONSTRUCT on the set of levels $U_1, U_2, \cdots, U_m$ as the one that permits the set U_i to reconstruct the secret only after the reconstruction of the secret by U_{i-1}. Now the set of levels with this RECONSTRUCT operator forms a chain. That is it is totally ordered set.

Example 3 *A set of three levels in LOAS* $U = \{U_1, U_2, U_3\}$ *is a strict POSET under the relationship operator RECONSTRUCT* $<$. *The set U forms a chain.* U_3 *is allowed to reconstruct the secret only after* U_2 *has reconstructed the partial secret; in turn* U_2 *is allowed to reconstruct the partial secret only after* U_1 *has reconstructed the partial secret.*

So the **Antichains of a POSET define generalized access structures; whereas the chains of a POSET define Level Ordered access structures**.

Dileep Kumar Pattipati received BTech from Jawaharlal Nehru Technological University, Hyderabad and M.Tech from University of Hyderabad, Hyderabad. Currently he is pursuing his PhD in Computer Science from IIT Madras, Chennai. He has software industry experience of 3 years. His research interests include Cryptography, Algorithms, and Semantic web.

Appala Naidu Tentu is a Research Scientist at CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus, Hyderabad. He obtained his M.Tech in Computer Science from National Institute of Technology, Suratkal (NITK), Karnataka, in 2010 and M.Sc from Andhra University, Visakhapatnam, in 2007. Currently, he is pursuing his PhD in Computer Science from JNTU Hyderabad. His research interests are in the areas of cryptography, cryptanalysis and design of security protocols.

V. Ch. Venkaiah obtained his PhD in 1988 from the Indian Institute of Science (IISc), Bangalore in the area of scientific computing. He worked for several organisations including the Central Research Laboratory of Bharat Electronics, Tata Elxsi India Pvt. Ltd., Motorola India Electronics Limited, all in Bangalore. He then moved onto academics and served IIT, Delhi, IIIT, Hyderabad, and C R Rao Advanced Institute of Mathematics, Statistics, and Computer Science. He is currently serving the Hyderabad Central University. He is a avid researcher. He designed algorithms for linear programming, subspace rotation and direction of arrival estimation, graph colouring, matrix symmetriser, integer factorisation, cryptography, knapsack problem, etc.

Allam Appa Rao is a Director at CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus, Hyderabad. He was the first to receive Ph.D from Andhra University in Computer Engineering in the year 1984. During his more than four decades of professional experience, such as First Vice Chancellor, JNTUK, Kakinada, A.P, Principal, College of Engineering (Autonomous), Andhra University. He shared his wisdom with fellow engineers and scientists across the globe through his innumerable research papers published in international journals and international conference proceedings. Indian Science Congress Association (ISCA) conferred him with "Srinivas Ramanujan Birth Centenary Award" Gold medal for his significant and life time contribution to the development of Science and Technology in the country specifically in the area of Computational Biology, Software Engineering and Network Security.

The Research on File Encryption Method Based on File Content Partitioning Restructuring

Hui Xiao[1], Hongbin Wang[1], and Meitong Lin[2]
(Corresponding author: Hongbin Wang)
Faculty of Information Engineering and Automation, Kunming University of Science and Technology[1]
No. 727, Jingming South Road, Chenggong New Area, Kunming, Yunnan 650504, P.R. China
School of Computer Science and Technology, Changchun University of Science and Technology[2]
No. 7089, Weixing Road, Changchun, Jilin 130022, P.R. China
(Email: whbin2007@126.com)
(Received June 23, 2015; revised and accepted Sept. 29 & Nov. 2, 2015)

Abstract

With the development of information technology and the application of information technology jumped into popularity, the electronic document was used as a form of information transmission and data storage, more and more electronic documents are facing the risk of being illegally acquired and viewed. In order to solve this problem, this paper proposes a novel file content protection method, which based on the file content partitioning restructuring. This method strengthens the file content protection and improves the documents security. But it also greatly increases the speed of encryption and decryption process.

Keywords: File encryption, file restructuring, files partitioning, information security

1 Introduction

In today's world, the security of information is associated with valid and reliable encryption algorithms [1]. With the popularity of the network, the individual or company's important files were potential threat (such as illegal steal, view, etc.). Therefore, we need to strengthen the Sensitive information files and important documents secrecy. However, choose what kind of encryption algorithm used to encrypt the file is a problem, and we hope that the algorithm is difficult to break, while the encryption and decryption process is fast enough, do not delay too much time. According to different situation to take different approach; they can better guarantee the important file content security.

Nowadays, there have two types of commonly used encryption algorithms at home and abroad, they are symmetric encryption algorithm and asymmetric encryption algorithm [6]. The DES algorithm is the old typical representative of symmetric encryption algorithm [14], the security of the encrypted file depends on the length of the password. If the password is not enough length, the encrypted file is easiest to crack by exhaustive method. If we build specialized hardware to crack the DES algorithm encryption file, the less time was being required, in addition, the DES algorithm need to 16 rounds of password replacement and substitution operation, then operate the documents again, it is need to consume a lot of time [5]. Another typical symmetric encryption algorithms like AES algorithm [4], a block encryption standard adopted by the US federal government, this standard is used to replace the original DES algorithm. However, the block length of AES algorithm is fixed at 128 bits, the key length needed is 128, 192 or 256 bits, has many limitations. Moreover, the greater of the used password length, the longer of the encryption and decryption process time-consuming.

The RSA algorithm is a typical representative of the asymmetric encryption algorithm [13], its safety is high, but this algorithm has more calculation. Encryption or decryption large files consumed time is hundreds of times of the AES algorithm, and the security of RSA algorithm depends on the large number decomposition, due to its public key is known, the private key can be calculated according to the public key. If we can find a kind of efficient large number decomposition algorithm, so it is not difficult to break out of the private key [7, 8].

In addition, there are many novelty encryption methods, Hwang et al. [2] proposed a simple batch verifying multiple RSA digital signatures. Their scheme is efficient to reduce computation of verifying multiple RSA signatures. Dong [3] proposed to enhance threshold secret sharing schemes based on the Chinese remainder theorem (CRT) by incorporating the well-known RSA Cryptosystem. Chang et al. [11] proposed an improved version to make the RSA-based certificateless scheme stronger and more secure. Bao et al. [9] proposed a cryptanalysis and improvement of Hwang et al. proposed scheme. Wang [12] proposed a method with bind the executable file to en-

crypt files, its advantage is run directly executable file input password can view the file, its no need to repeat the encrypted file, but the program need to store the correct password to compare, namely the encrypted file contains the correct password. We also can obtain the password through other means. This method reduces the security of the encrypted file content. Wang et al. [10] proposed a policy based de-duplication architecture, using the mechanism of security proxy (SP) and random storage, which separate storage services and security services to ensure the security of user data and improve the system efficiency at the same time.

But above all, there is a password related information, encrypted file is always insecure, public key may be cracked; stored passwords also can be obtained. According to this problem, this paper present file encryption algorithm based on file content partitioning restructuring to strengthen the file content protection.

The rest of paper is organized as follows. Section 2 introduces the file content partitioning restructuring encryption method. Section 3 introduces the experiment and discussion. Finally, Section 4 concludes this paper.

2 File Content Partitioning Restructuring Encryption Method

2.1 File Encode Procedure

For any file content, when the computer process these documents content, the file content was converted to "0" and "1" binary byte code. If we make the large file encrypted in binary form, the file content encryption degree equivalent to a file "0" and "1" chaos degree. In order to increase a file "0" and "1" chaos degree, we can use a variety of complex algorithms to encrypt the file content, such as exclusive or, modular operation, etc., but it does not necessarily have a high level of encryption, because in the encryption process will inevitably make some useless. For example, a third-order Rubiks cube, American scientists have proved that any disturb the Rubiks cube can be restored within twenty steps, it shows that how many times disturb the Rubiks cube, the Rubiks cube can be restored within twenty steps. The encryption file decoding is just like the revivification Rubiks cube, it through a variety of complex algorithms dealing with ten times or even dozens of times. Maybe it only need a few steps can be restored the file content.

By this we know that as long as the method is reasonable and effective, even through a simple encryption process, the level of encryption may be also complex. Therefore, this paper considers the encryption algorithm based on file content partitioning restructuring, the main idea is: according to the user input password, the passwords ASCII code as password in encoding process. We make the file content divided into the passwords length pieces (such as a passwords length is 20, then the file content is divided into 20 pieces), and each small pieces with the corresponding password is proportional to the size (such as user input "lThy9686", the program reads ASCII code "1088410412157545654" as a password, then according to user input passwords each character ASCII value divided file content into 108, 84, 104, 84, 57, 54, 56, 54 pieces), then according to the sorting rule make the partitioning file content integrate into a new file, the sorted new file is a restructured file. So that we can use ASCII international generality to increase the security of encryption algorithm and make the encryption algorithm has scope of application. If the user wants to encrypt files binding the users computer (that is, the encrypt files only allowed decoding on users computer), you can choose the computer's MAC address or hard disk serial number as part of the password in the encryption process, thus someone stealing encoded files to other computers cannot be decoded.

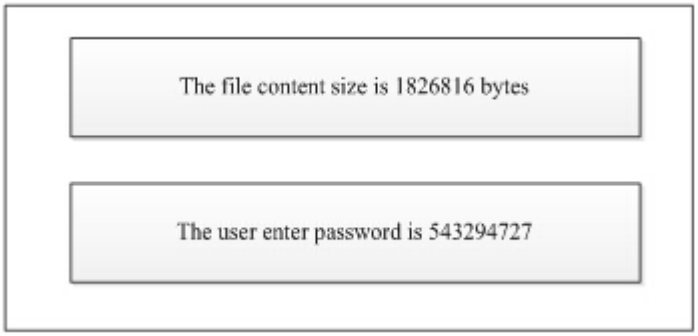


Figure 1: The file content size and password

2.2 File Encode Procedure

If you have a 1.74 MB file, we want to encrypt this file based on file content partitioning restructuring (we make file in bytes as the smallest unit to avoid the numerical too big to inconvenience description, in practical cases the smallest unit is bit.). The users need to enter a password first, for example the password is 543294727, and it is converted to ASCII as 53-52-51-50-57-52-55-50-55, in order to make the block proportion has distinct distinction, we use the user enter password numbers to divide the file content. The file content size and password as shown in Figure 1.

We give the input password a sequence number and sort password from small to big order. At the same time, if the password sequence changed, the sequence number will also change. The password sorting is shown in Figure 2.

The file content is divided into nine blocks, which is proportional to the size of password; you can see different password numbers have difference size of file block. The file block sorting is shown in Figure 3.

If we want to obtain each block file content size, we only need sum of the password numbers, then the file content size divided by the sum of the password numbers and can get the unit block size (if any remainder can choose in the

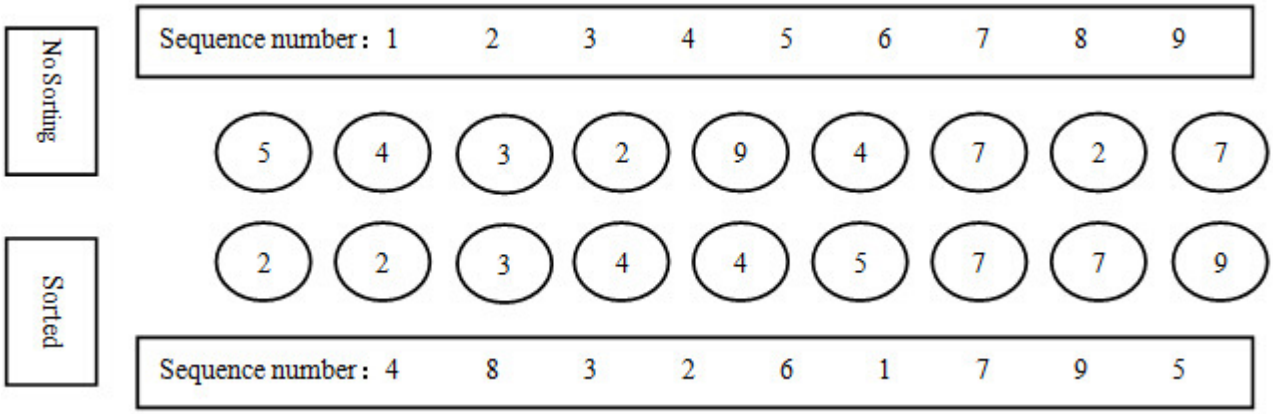


Figure 2: The password sorting

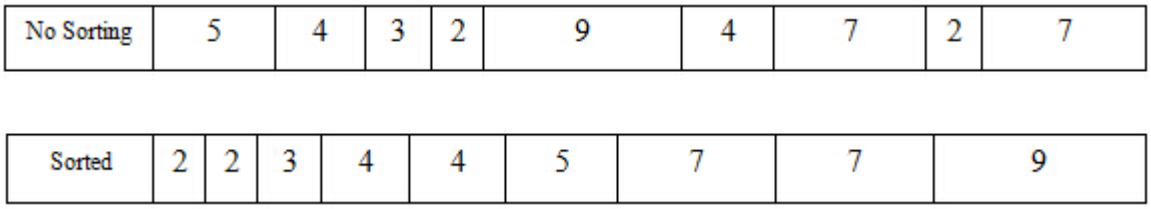


Figure 3: File block sorting

last block), such as Equation (1):

$$1826816 \div (5+4+3+2+9+4+7+2+7) = 42484 \cdots 4. \quad (1)$$

By Equation (1), the unit block size is 42484 bytes (the remainder is 4), each password corresponding block size is shown in Table 1.

We can see from Table 1, the password corresponding block is proportional to the size (the last password $7's$ file size is more than the first password $7's$ file size 4 bytes, because of containing the remainder 4 bytes), the node column denote read each file blocks end position (we assume the head node location is 0, the password $5's$ position is 0-212420, the password $4's$ position is 212420-382356, and so on).

The encrypted file is the original file content partitioning restructuring. The sorting process of the file content partitioning is stabilizing order; the same size partitioning does not change the original order. Its size of each partitioning is shown in Table 2.

We can see from Table 2, the node column denote write each file $block's$ end position. We assume the encrypted $file's$ header node location is 0, the first password $2's$ file content is copying the original file content from 509808 to 594776, and the second password $2's$ file content is copying the original file content from 1444456 to 1529424, and so on. Such copies of each block and writing into new file, the new file is encrypted files, the process is the original file content partitioning restructuring.

However, this is just a simple example. In the actual situation, we will judge each block. If one block is too big, then encrypt this block by using the same password, and recursion continues until all blocks are small enough.

2.3 File Decode Procedure

The decoding process and the encoding process in the same way, it is sorting operation on file content. Most of operations and encryption process are exactly the same, just order change.

Likewise, the decoding process also requires the user to enter the password; we can refer to Figure 1. the program will rank password refer to Figure 2.The only difference is the encoding process is the inverse process of decoding, according to Table 1 and Table 2, the encryption process is the original documents according to Table 1 read step by step, according to Table 2 written to the file step by step; the encrypted file decoding process is according to the Table 2 read step by step, according to Table 1 written to the file step by step, the process of encoding and decoding depends on the sequence of Table 1 and Table 2.

3 Experiment and Discussions

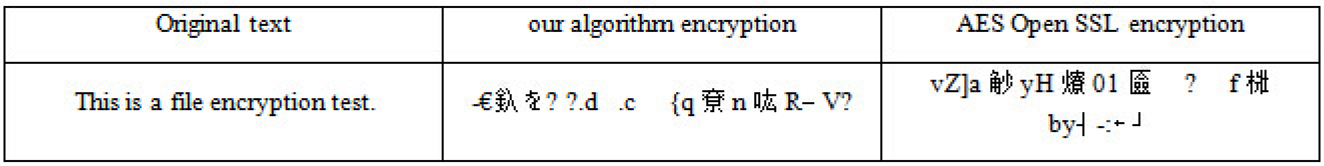
We use our proposed algorithm and AES algorithm to encrypt the same file and then test the effectiveness of our proposed encryption algorithm. The test document content is "This is a file encryption test.", and encrypt

Table 1: The size of original file block

Password	5	4	3	2	9	4	7	2	7
File size	212420	169936	127452	84968	382356	169936	297388	84968	297392
Termination node	212420	382356	509808	594776	977132	1147068	1444456	1529424	1826816

Table 2: The size of encrypted file block

Password sorting	2	2	3	4	4	5	7	7	9
File size	84968	84968	127452	169936	169936	212420	297388	297392	382356
Termination node	84968	169936	297388	467324	637260	849680	1147068	1444460	1826816



Original text	our algorithm encryption	AES Open SSL encryption
This is a file encryption test.	-€釻 を? ?.d .c {q 寳 n 呟 R– V?	vZ]a 䖬 yH 燎 01 匳 ? f 栿 by┤ -:← ┘

Figure 4: File block sorting

results are shown in Figure 4.

We use 10 files to verify the performance of these encryption methods, and use the same 128 bits password as EncryptFilesTest. The parameters for computer is AMD A4 CPU 1.5 GHz and RAM 6GB, the file sizeencryption time cost and decryption time cost are shown in Figure 5.

We can be seen from Figure 5, with the increase of encrypted file contents, our method on the encryption and decode time consumption is better than the AES algorithm. The experimental results verify our algorithm efficiency. The file encryption based on the file content partitioning restructuring, which encryption process is read and write the file process. Just only read the order and position is different, namely the cloning process is the encrypt files, so that we can maximize the speed of encryption process. At the same time, the file content only needs a simple calculation before file encryption (namely, we need to compute the size of each file content block and end location), encryption process does not occupy too much time.The execution speed of these two encryption algorithms based on encrypt and decrypt the biggest file as shown in Table 3.

By encoding/decoding process, the file itself and the program will not store any information related to the password, the password is only used as a keyword, namely the user enter a password would be to encrypt or decrypt files once, the program will not compare password correctly or not, it also ensures that no one except the user know the correct password, don't leave any information may calculate the password.

Our encryption method also made the exhaustive method decrypt file changed unrealistic and doesn't like other encryption algorithm. It use the password in any length (the length greater than 0). Due to the algorithm would recursion encrypt file until all blocks are small enough, it is possible to provide the same security for different lengths of password, but the encryption time is likely to change. In this case, you do not know the password is 128 bits or 1024 bits, it is not a fixed size, and it may be an odd number of bytes such as 17 or 23. Sometimes you have to decrypt the file completely then you could know the password is correct or not, instead of just compare password. According to the results above, a 600MB file decryption will consume nearly a minute of time, a nine digit password has a billion of possibilities, the exhaustive method to crack the code will be consumed more than 1900 years.

Thus if the user input the correct password decrypt documents will not cost too much time by our proposed method, the decryption speed always depends on the hard disk read and write speed, but the exhaustive method would increase cost observably.

4 Conclusions

In this paper, we proposed an efficient method for file content based on the file content partitioning restructuring. So we can use different sorting ways to enhance the encryption algorithm performance, the encryption algorithm is still has insufficient place, such as the large file encryption need cost slightly longer times, encryption require larger memory space. We will continue to improve our encryption method in the future work.

File size	Number of bytes	Time cost (second)			
		our algorithm		AES Open SSL	
		Encryption	Decryption	Encryption	Decryption
2.63KB	2,696	0.001	0.001	0.001	0.001
256KB	262,656	0.016	0.015	0.031	0.031
1.62MB	1,707,590	0.421	0.187	0.249	0.124
6.72MB	7,057,203	1.591	0.405	0.999	0.578
44.5MB	46,743,761	4.649	2.371	4.274	3.761
107MB	112,895,950	8.985	5.741	10.172	9.063
616MB	646,462,820	54.389	32.636	66.116	52.104
1.09GB	1,174,199,531	96.341	59.751	119.045	95.417
1.58GB	1,703,508,231	144.767	87.974	187.816	139.231
1.97GB	2,121,837,981	204.319	119.326	219.324	174.675

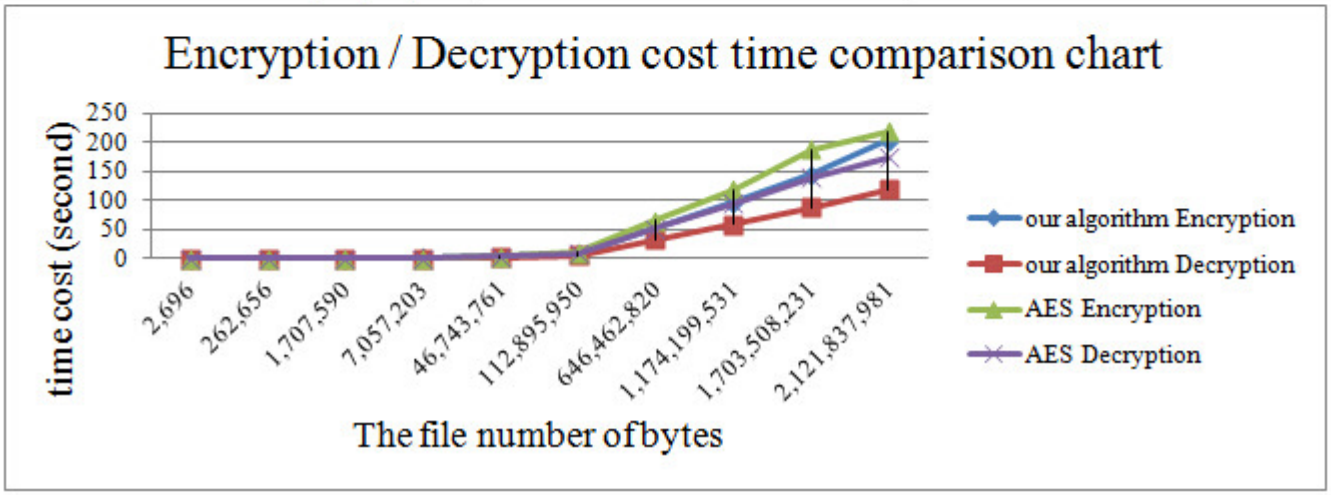


Figure 5: The encryption/decryption cost time comparison chart

Table 3: The execution speed

Execution speed (MB/s)	**our algorithm**	**AES Open SSL**
Encryption	9.9	9.2
Decryption	16.9	11.6

Acknowledgments

This work is supported by the National Nature Science Foundation of China (61462054); the Science and Technology Plan Projects of Yunnan province (2015FB135); the Key Project of Yunnan province Education Department (2014Z021); the Nature Science Research Foundation of Kunming University of Science and Technology (KKSY201403028).

References

[1] M. Babaei, "A novel text and image encryption method based on chaos theory and dna computing," *Natural Computing*, vol. 12, no. 1, pp. 101–107, 2013.

[2] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for rsa cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[3] 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.

[4] M. X. He and H. Lin, "Implementation of the advanced encryption standard (aes)," *Application Research of Computers*, vol. 12, no. 0, pp. 61–63, 2002.

[5] M. Hu and R. Lu, "Analysis and research of the security of DES algorithm," *Journal of Inner Mongolia University (natural science edition)*, vol. 36, no. 6, pp. 693–697, 2005.

[6] M. S. Hwang and I. C. Lin, *Introdution to Information and Network Security (in Chinese)*, 5ed., McGraw-Hill, 2014.

[7] Y. Jin, X. Cui, N. Jiang, "Design of file encryption system based on RSA algorithm," *Journal of Dalian Nationalities University*, vol. 15, no. 5, pp. 535–538, 2013.

[8] Y. Kou, J. Tian and C. Chen, "Research and simulation of timing attacks on rsa," *Computer Technology and Development*, vol. 20, no. 8, pp. 150–158, 2010.

[9] C. C. Lee, F. Bao and M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple rsa digital signatures," *Applied Mathematics and Computation*, vol. 172, no. 0, pp. 1195–1200, 2006.

[10] Y. Lu, Z. Y. Wang and G. Z. Sun, "A policy-based de-duplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[11] C. Y. Sun, C. C. Chang and S. C. Chang, "A strong RSA-based and certicateless-based signature scheme," *International Journal of Network Security*, vol. 18, no. 2, pp. 201–208, 2016.

[12] X. Wang, "Based on the bundled file encryption technology and implementation," *Journal of TaiYuan Urban Vocational College*, vol. 12, pp. 135–136, 2012.

[13] G. Zhang, H. Wang and Z. Li, "Public key cryptography RSA algorithm," *Information Technology*, vol. 0, no. 8, pp. 34–36, 2011.

[14] J. Zhang and L. J. Zhu, "The analysis and realization of arithmetic of des encryption," *Software Guide*, vol. 3, no. 36, pp. 95–97, 2007.

Hui Xiao is a college undergraduates student. His current study in Faculty of Information Engineering and Automation from Kunming University of Science and Technology. His major is computer science and technology.

Hongbin Wang received his Ph.D in computer science in 2013 from the Jilin University, Changchun,China. He is a lecturer in the Faculty of Information Engineering and Automation from Kunming University of Science and Technology. His current research interests include intelligent information system, natural language processing and network security.

Meitong Lin is a college undergraduates student. Her current study in School of Computer Science and Technology from Changchun University of Science and Technology. Her major is computer science and technology.

A Secure and Robust Certificateless Public Key Steganography Based on SVD-DDWT

Osman Wahballa, Abubaker Wahaballa, Fagen Li and Chunxiang Xu
(Corresponding author: Abubaker Wahaballa)

School of Computer Science and Engineering, University of Electronic Science and Technology of China
No.4, Section 2, North Jianshe Road, Chengdu, P.R. China
(Email: wahaballah@hotmail.com)
(Received July 15, 2015; revised and accepted Sept. 29 & Nov. 2, 2015)

Abstract

Security and undetectability are main goals of steganographic systems. This paper proposes a novel certificateless public key steganography that allows two parties that have no prior knowledge of each other to communicate covertly over public channel. Firstly, secure and high efficient rate of key distribution are provided. Secondly, proper stego and destego are introduced based on Distributed Discrete Wavelet Transform (DDWT) and Singular Value Decomposition (SVD).Thirdly, we present the Matlab analysis of the original and stego images, which proves the robustness of our scheme. Finally, the analyses demonstrate that our scheme meets all security requirements of steganographic system and resists various kinds of sophisticated attacks.

Keywords: Certificateless public key steganography, distributed discrete wavelet transform, singular value decomposition

1 Introduction

While cryptography is about enciphering the content of messages in secret code or cipher, steganography aims to transmit the content of messages inside a perfectly innocent covers. Steganography [13] is a skill of concealing communication between two parties in the presence of third party called adversary. The term derived from Greek, literally means hidden writing. It includes many different forms of secret communication techniques that hide a secret message within *cover-text* so that others cannot see or know of any hidden message. These techniques have evolved from a simple and primitive techniques, such as invisible inks and microdots to other, more complex and sophisticated, such as covert channels, spread spectrum, and transformation domain techniques.

In order to safeguard information and communication between sender and receiver, and to stave off an attacker from breach of sensitive information, a steganographic message will appear to be something else as shown in Figure 1. It can be: plain text, image, an audio, video or TCP/IP [25].

Current public key steganography schemes have been constructed based on traditional public-key infrastructure (PKI) or Identity-based cryptosystem (IBC). However, PKI-based schemes are adversely affected by the complex procedures of certificates management and verification, while the obvious drawback of IBC is an escrow problem.

With a view to solve the key escrow problem in identity-based public key cryptosystem (ID-PKC) [7], Al-Riyami and Paterson [3] proposed a certificateless public key cryptosystem (CL-PKC) which contains the attractive features of ID-PKC (certificateless property). Furthermore, the reliance on trusted third party is much reduced. Wang *et al.* [26] and Baek *et al.* [6], both made their marks to list the features of CL-PKC which include:

- CL-PCK facilitates the complex certificate management process in the traditional public key cryptography;
- The key generation center (KGC) in CL-PKC is incapable to generate the user's whole private key, which does not have the highest priority for key generation.
- CL-PCK provides lower computational costs and communication overheads.

Finally, we remark that CL-PKC provides several useful and appealing features. Therefore, we take advantage of these features to construct a secure and robust certificateless public key steganography scheme.

1.1 Motivations

The *Prisoner's Problem* [21] is considered as the motivation of this paper. In this problem, Alice and Bob are in prison, and are considering a means to escape but the only way they can relay information to and from each other is via a public channel under the hearing and eyesight of

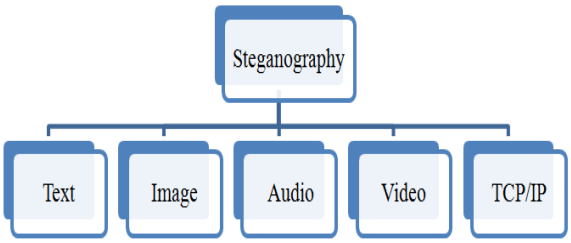


Figure 1: Types of steganography

a warden, Wendy. With a view to avoid Wendy's suspicion, they have to communicate as invisible as possible so that they will not be revealed by Wendy. Invisibility is an essential point in steganographic systems. Furthermore, the efficient key management of CL-PKC is useful for public key steganographic systems, especially when the Key Generation Center (KGC) is incapable to hold the user's whole private key, which does not have the full power for key generation. It just generates a user's partial private key from the user's identity. That is why certificates are no longer needed in CL-PKC.

Our contributions are the following folds:

1) A novel public key certificateless steganography is proposed;
2) Highly efficient rate for key distribution and management is provided;
3) The key escrow problem is addressed.
4) Proper *Stego* and *Destego* are offered.

1.2 Security Issues in Steganographic Systems

In general, a practical and secure steganographic system should satisfy the following requirements:

- ***Robustness***: The embedded data must be kept intact if the stego-system undergoes transformation, such as spatial domain and frequency domain transformation; linear and non-linear filtering; addition of random noise etc.
- ***Undetectability***: The hidden text (steganographic message) should appear identical to all possible statistical tests which can be carried out.
- ***Indistinguishability***: It means that it is hard to distinguish between covertext and stegotext.
- ***Security***: It is said that the embedded algorithm is secure if the hidden-data is not subject to removal after being discovered by the attacker.

The paper is organized as follows. Related works and pervious result are discussed in Section 2. In section 3, we briefly introduce the preliminaries used in this paper. A novel certificateless steganography is proposed in section 4. Section 5 deals with efficiency comparison and security analysis. The experimental results are presented in Section 6. Finally, conclusion and recommendation for future works are given in Section 7.

2 Related Works

Public key steganography was first considered by Anderson [4]. However, only informal security model was proposed. In 2002, Guillon *et al.* [9] introduced an experimental study for steganalysis of scalar costa scheme (SCS). This scheme was applied to PCM audio contents. It was designed based classical public-key cryptosystem that is RSA. The disadvantages of RSA are: i) Very slow key generation; ii) Two-part key is vulnerable to GCD attack if poorly implemented. In 2004, the basic notations of steganographic security against adaptive chosen-covertext attacks (SS-CCA) and steganographic security against publicly-detectable replayable adaptive chosen-covertext attacks (SS-PDR-CCA) was defined formally by Backes and Cachin [5] in IBM laboratory at Zurich. Ahn and Hopper [2] introduced the first protocols for public-key steganography and steganographic key exchange in random oracle model. Le and Kurosawa [14, 15, 16] proposed serial versions of stegosystem. However, these schemes are not in line with the standard model of chosen hiddentext attacks. Hopper and Ahn [12] proposed a provably secure steganography scheme based on unbiased functions. However, this scheme had extremely low information rates. Ahadpour *et al.* [1] proposed a method for the public key steganography based on Discrete Cross-Coupled Chaotic Maps. This method was used to specify the location of the different parts of the secret data in the JPEG image. Ahadpour's method was based on the diffie-hellman key exchange algorithm. However, there are some drawbacks in this algorithm that are discrete logarithm and Man-in-the-Middle attack. Recently, Ruffing *et al.* [19] introduced the concept of *Identity-Based Steganography*. However, the key escrow problem in this scheme is not a good property for public key steganographic systems.

In this paper, an efficient certificateless public key steganography scheme is proposed. Our model does not only satisfy the security requirements of steganographic systems, but it is also able to improve the computational costs and communication overheads.

3 Preliminaries

In this section, we give a brief introduction on the preliminaries required in this paper which include computational hardness assumptions, discrete wavelet transform and singular-value decomposition (SVD).

3.1 Computational Hardness Assumption

Our scheme is based on the hardness assumptions as follows:

1) Discrete Logarithm (DL) Problem: Given a generator P of a cyclic group $\mathbb{G}^*$ with order q, and $x \in \mathbb{Z}_q^*$ satisfying $Q = xP$.

2) Divisible computational DiffieHellman (DCDH) problem: Given (aP,bP) then compute $ab^{-1}P$, where $P \in \mathbb{G}$ is the generator and $a, b \in \mathbb{Z}_q^*$ is unknown.

3.2 Discrete Wavelet Transform

In the last few decades, Discrete Wavelet Transform (DWT) [8, 20] had been adopted and deployed in an extensive range of applications including numerical analysis, signal analysis, pattern recognition, computer vision, image/video coding, steganography and watermarking. Wavelet transform provides both time and frequency information simultaneously. In this transform, time domain is passed through low-pass and high-pass filters (band-pass) to get low and high frequencies respectively. The advantage of DWT is that provided a better compression ratio without losing more information of image.

Discrete wavelet transform has several types. The oldest and most known one is the Haar DWT [22] which includes two steps, namely the horizontal process and vertical process. The neighboring pixies is used to perform the horizontal process from left to right then execute the vertical process from top to bottom as shown in Figure 5. The LL sub-band is used to embed the steganographic message (secret message). However, this sub-band is vulnerable to the image cropping attacks. In order to address this problem, Lin *et al.* [17] suggested Distributed Discrete Wavelet Transform (DDWT). In this method, multi-scale DDWT is used to transform the image data from spatial domain into frequency domain and then hide the steganographic message in the frequency domain and perform inverse multi-scale DDWT transformation (IDDWT) to get stego image in spatial domain. The steganographic method in this paper is based on Lin's Distributed Discrete Wavelet Transform (DDWT).

3.3 Singular-Value Decomposition

Singular-Value-Decomposition (SVD) is a useful tool for matrix factorization [23]. For any digital image A of size $m \times n$ with $m \geq n$, can be represented by A's SVD as follows:

$$A = U\Sigma V^T = \sum_{i}^{m} \sigma_i u_i v_i^T \quad (1)$$

where $U_{m\times t}$ and $V_{n\times t}$ are orthogonal matrices and $\Sigma_{t\times t}$ is a diagonal matrix representing the singular values on

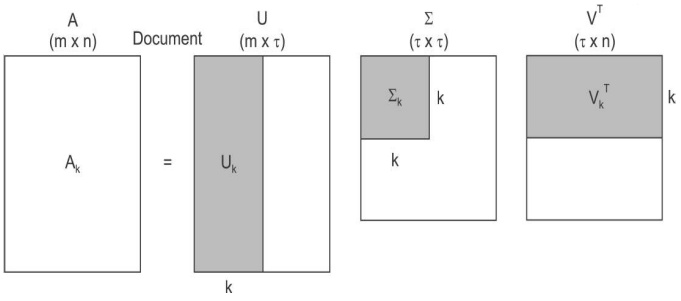


Figure 2: SVD decomposition

top of $m - n$ rows of zeros:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & 0 \\ & \sigma_2 & & & \\ & & .. & & \\ & & & .. & \\ 0 & & & & \sigma_m \end{bmatrix}$$

The columns of matrix U are the left singular vectors (eigenvectors U_k) and V^T has rows that are the right singular vectors (eigenvectors V_k). As shown in Figure 2, calculating the SVD consists of finding the eigenvectors and eigenvalues of U_k and V_k^T.

4 Proposed Model

In this section a novel public key certificateless steganography is proposed. The notations of our proposed model used in this paper are shown in Table 1. The KGC is adopted as a key generation center. Our proposed model is expressed diagrammatically in Figure 3. Alice and Bob are in prison, and want to relay information to and from each other is via a public channel under the watch of a warden, Wendy. To avoid Wendy's censorship, Alice sends to Bob some innocuous contents. Alice is said to be active when she embeds a hidden message h_{txt} modifying the cover-text C_{txt} into stego-text S_{txt}. Alice is not active when she sends really innocuous contents.

In order to establish a secure communication channel between Alice and Bob, we describe the eight algorithms needed to define our scheme based on Alriyami and Paterson [3] and He *et al.* [10] schemes, which include: `Setup`, `Set Secret Value`, `Partial Private Key Extract`, `Set Private key`, `Set Public Key`, `Key-Agreement`, `Stego` and `Destego`.

1) `Setup`: Initially, the KGC inputs the security parameters. These include the tuple $\{F_q, E|F_q, G, P\}$ as defined in Section 3. The KGC randomly chooses its master-key $s \in \mathbb{Z}_n^*$ and computes its public master-key $P_{pub} = sP$, and chooses two hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_n$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_n$.

   Finally, the KGC publishes the system parameters: $params = (F_q, E|F_q, G, P, P_{pub}, H_1, H_2)$.

2) `Set Secret Value`: Alice A with identity ID_A selects $x_A \in \mathbb{Z}_n^*$ and sets x_A as secret value.

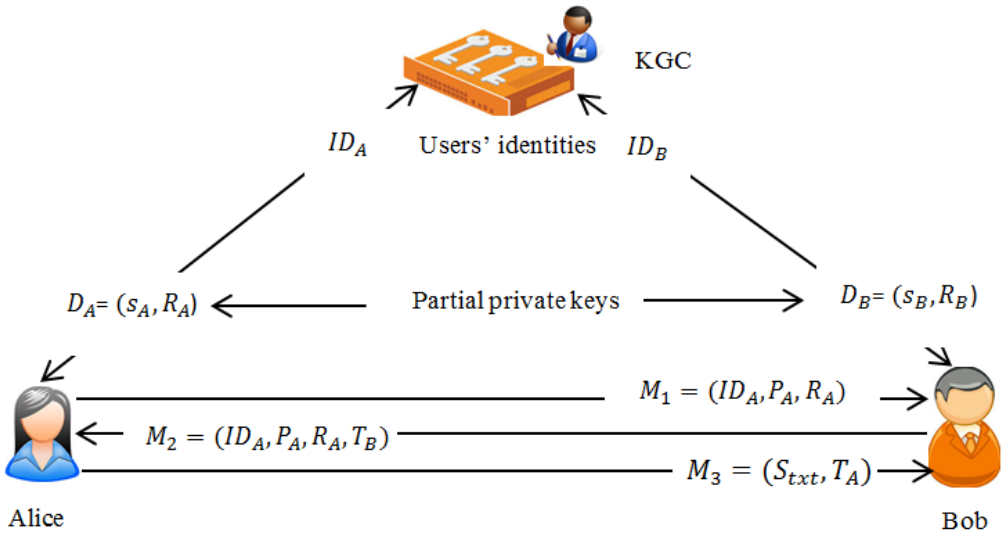


Figure 3: Proposed model

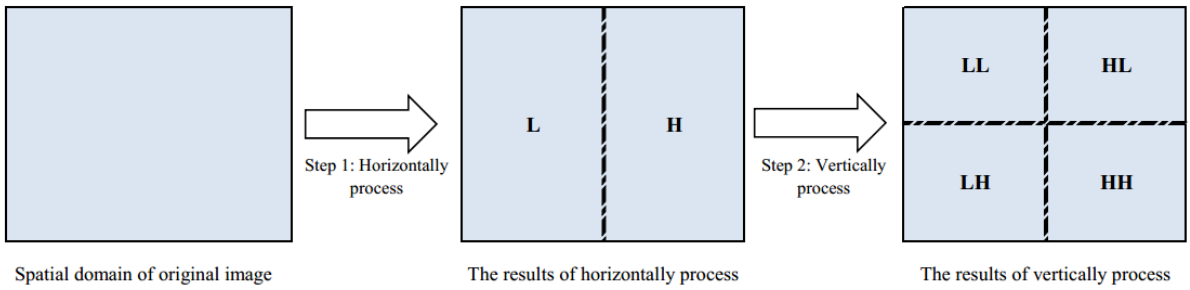


Figure 4: 1-scale DDWT

3) `Partial Private Key Extract`: KGC computes the partial private key of Alice with identity ID_A as follows:
   - KGC chooses $r_A \in \mathbb{Z}_n^*$, computes: $R_A = r_A P$ and $h_A = H_1(ID_A, R_A)$;
   - Then, KGC computes $s_A = r_A + h_A s \bmod n$;
   - KGC sets the tuple $D_A = (s_A, R_A)$ as partial private key.
   - KGC sends D_A secretly to Alice.

4) `Set Private key`: When Alice receives D_A from the KGC, Alice can validate the partial private key by checking whether the equation $s_A P = R_A + h_A P_{pub}$ holds. If it holds, then Alice sets the pair $S_A = (x_A, D_A)$ as her full private key.

5) `Set Public Key`: Alice computes her public key as $P_A = x_A P$:

   Bob with identity ID_B can repeat algorithms from 2 to 5 to generate his keys.

6) `Key-Agreement`: The common authenticated per session secret key can be computed at both sides as follow:
   - Alice sends $M_1 = (ID_A, R_A, P_A)$ to Bob;
   - Upon Bob receiving M_1, he chooses at random the ephemeral key $b \in \mathbb{Z}_n^*$ and computes $T_B = b(P_A + R_A + H_1(ID_A, R_A)P_{pub})$. Then, Bob sends $M_2 = (ID_B, R_B, P_B, T_B)$ to Alice;
   - After receiving M_2, Alice chooses at random the ephemeral key $a \in \mathbb{Z}_n^*$ and computes $T_A = a(P_B + R_B + H_1(ID_B, R_B)P_{pub})$. Then, Alice sends $M_3 = (T_A)$ to Bob;
   - Then, both sides can compute the shared secrets as follows:
     - Alice computes $K_{AB}^1 = (x_A + s_A)^{-1} T_B + aP$ and $K_{AB}^2 = a(x_A + s_A)^{-1} T_B$;
     - Bob computes $K_{BA}^1 = (x_B + s_B)^{-1} T_A + bP$ and $K_{BA}^2 = b(x_B + s_B)^{-1} T_A$.
   - Eventually, Alice and Bob can compute the shared secret keys as:

$$\begin{aligned} sk &= H_2(ID_A||ID_B||T_A||T_B||K_{AB}^1||K_{AB}^2) \\ &= H_2(ID_A||ID_B||T_A||T_B||K_{BA}^1||K_{BA}^2). \end{aligned}$$

7) `Stego`: If Alice want to send secret message h_{txt} (hidden message) to Bob into cover-content C_{txt}, she can execute the following algorithm:

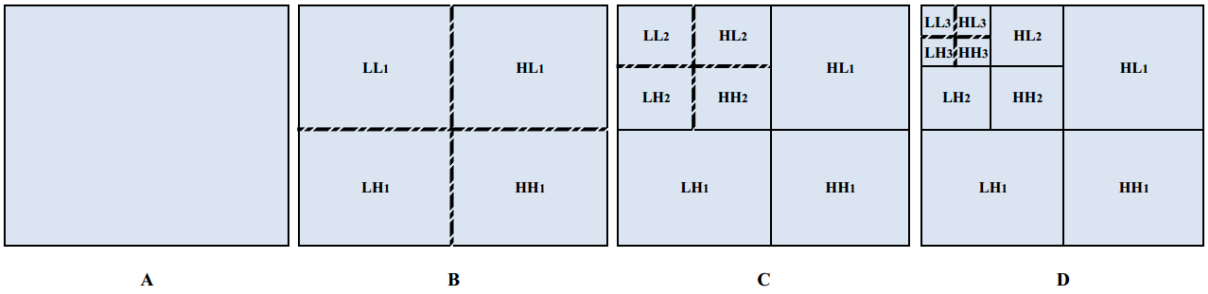


Figure 5: Multi-scale DDWT transforms: (A) The original image (B) 1-scale DDWT (C) 2-scale DDWT (D) 3-scale DDWT

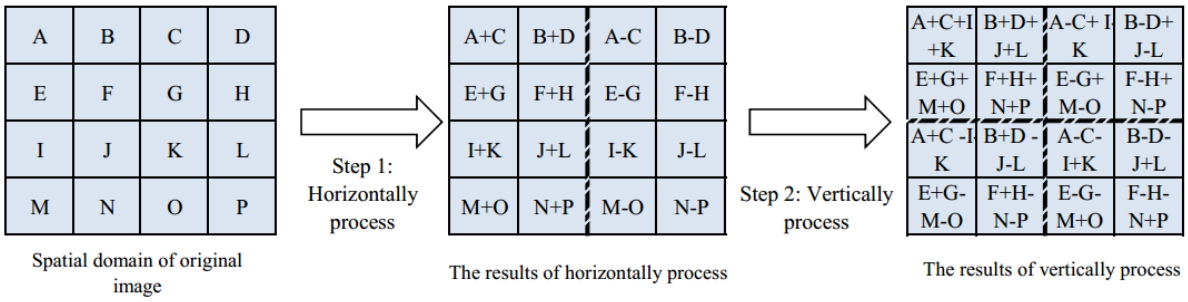


Figure 6: 1-scale DDWT: Horizontal and vertical processes on an original image with 4×4 dimensions

- Alice embeds a secret message h_{txt} into stego-content S_{txt} by modifying the cover-content C_{txt} as: $S_{txt} = \varepsilon_{sk}(h_{txt}, C_{txt})$, where ε is the embedding algorithm;
- Then, Alice sends the S_{txt} to Bob.

8) `Destego`: Bob destegos Alice's hidden message with shared key as follows: $< h_{txt}, C_{txt} >= \beta_{sk}(S_{txt})$.

4.1 Steganographic Method

The steganographic method in this paper is based on Lin's et al DDWT, which consists of two steps: horizontal process and vertical process. The details of these processes are described as follows.

4.1.1 Horizontal Process

In this process, the original image is separated horizontally into two equal blocks. Then, from left to right add and subtract corresponding pixels on the two sub-blocks. At the end of this process, the pixels on the left sub-block are replaced with result of addition, while the result of the subtraction replaces the pixels on the right sub-block. The left sub-block represents the low frequency domain and is denoted as L; the right sub-block represents the high frequency domain and is denoted as H.

4.1.2 Vertical Process

From the blocks generated by the horizontal process above, the image is separated vertically into equal sub-blocks. Then, from upper to lower add and subtract corresponding pixels on the two sub-blocks. The pixels on the upper sub-block are replaced with result of addition, while the result of the subtraction replaces the pixels on the lower sub-block. Thus, four sub-blocks are generated and denoted as LL, LH, HL and HH. Figure 4 shows these sub-blocks. The Step-1(horizontal process) and Step-2 (vertical process) are repeated 2 k times. Figure 5 shows the k-scale DDWT transform, while 1-scale DDWT: horizontal and vertical processes on an original image with 4×4 pixels is shown in Figure 6.

As indicated in embedding algorithm ε, we set input, output and the algorithm parameters in Steps 1-3. Steps 4-10 show the decomposition process (multi-scale DDWT) for the coverImg. Then, we perform SVD for diagonalImg blocks. Steps 12-16 show the embedding process. Finally, we apply inverse DDWT for srego-blocks to obtain the stego-image.

As shown in Extract Algorithm β, from Steps 1-3 we set the input, output, and the parameters. Then, we perform DDWT for stegoImg in Steps 4-9. The SVD process is applied in Step 10, while we extract the secret message Msg in Steps 11-15.

5 Efficiency Comparison and Security Analysis

The security and efficiency of the proposed scheme is analyzed in this section. Security requirements of our proposed model are discussed in Section 5.1, while efficiency

Algorithm 1: Embedding algorithm ε

Input: $coverImg, Msg$
Output: $stegoImg$

$Bit \leftarrow M_0, M_1, .., M_{65535}$ // Extract Bit set of Msg
$w \leftarrow coverImg.width$
$h \leftarrow coverImg.height$
// Decomposition process
for $i \leftarrow w$ ***downto*** 0 **do**
  for $j \leftarrow h$ ***downto*** 0 **do**
    $w \leftarrow (w+1)/2$
    $h \leftarrow (h+1)/2$
    $HorizontalImg^{LLLH} \leftarrow coverImg(w,h)$
    $VerticalImg^{HLHH} \leftarrow coverImg(w,h)$
    $diagonalImg^{LHHL} \leftarrow coverImg(w,h)$
    $[U,S,V] \leftarrow SVD(diagonalImg^{LHHL})$
      // Perform SVD for diagonalImg blocks
  for $k \leftarrow 1$ ***to*** $Msg.length$ **do**
    if $M_k = 0$ **then**
      $HorizontalImg_{ij}^{LL} \leftarrow M_k$
      else
        $VerticalImg_{ij}^{HH} \leftarrow M_k$
$stegoImg \leftarrow IDDWT(stego\_blocks)$// Apply Inverse DDWT for srego_blocks to obtain the stego-image
return $stegoImg$

Algorithm 2: Extract algorithm β

Input: $stegoImg$
Output: Msg

$Bit \leftarrow M_0, M_1, .., M_{65535}$ // Bit set of Msg
$w \leftarrow StegoImg.width$
$h \leftarrow StegoImg.height$
for $i \leftarrow w$ ***downto*** 0 **do**
  for $j \leftarrow h$ ***downto*** 0 **do**
    $w \leftarrow (w+1)/2$
    $h \leftarrow (h+1)/2$
    $StegoImg^{LLLH} = StegoImg(w,h)$
    $StegoImg^{HLHH} = StegoImg(w,h)$
    $[U,S,V] \leftarrow SVD(stegoImg^{LHHL})$
      // Perform SVD for diagonalImg blocks
  if $StegoImg_{ij}^{LLHH} < 0$ **then**
    $M_i = 0$;
    else
      $M_i = 1$;
  $Msg = Combine(M_i)$
return Msg

Table 1: Notations of our proposed model

Notation	Meaning
KGC	A key generation center
ID_A	Alice's A's Identity
ID_B	Bob's B's Identity
P_A	Alice's public key
P_B	Bob's public key
P_{pub}	The KGC's master key
X_A	Alice's secret value
X_B	Bob's secret value
S_A	Alice's private key
S_B	Bob's private key
D_A	Alice's partial private key
D_B	Bob's partial private key
sk	Shared secret key
h_{txt}	A hidden text (steganographic message)
C_{txt}	A cover content
S_{txt}	Stego content
H_1, H_2	Two hash functions
ε	Embedding algorithm (steganography algorithm)
β	Extract algorithm

comparison is presented in Section 5.3.

5.1 Correctness

It can be easily seen that $K_{AB}^1 = K_{BA}^1$ and $K_{AB}^2 = K_{BA}^2$. Hence, the shared secrets are agreed.

$$\begin{aligned} K_{AB}^1 &= (x_A + s_A)^{-1} T_B + aP \\ &= bP + aP \\ K_{BA}^1 &= (x_B + s_B)^{-1} T_A + bP \\ &= aP + bP \\ K_{AB}^2 &= a(x_A + s_A)^{-1} T_B \\ &= abP \\ K_{BA}^2 &= b(x_B + s_B)^{-1} T_A \\ &= baP. \end{aligned}$$

5.2 Security

As proved in [10], our protocol satisfies all security requirements of authenticated key agreement:

- *Known-key secrecy*: It allows to run the key exchange protocol several times. In each time, Alice and Bob should obtain a unique session key which depends on

Table 2: Efficiency comparison

Steganographic Model	Computational Costs			Message Exchange
	T_{mul}	T_H	T_e	
Ruffing *et al.* [19]	2	4	2	2
Our model	8	5	0	3

every particular ephemeral key $a, b \in \mathbb{Z}_n^*$ for Alice and Bob receptively. Even if the adversary $\mathcal{A}$ has learned some other session keys, s/he cannot compute the keying point $E_q(a, b)$ from aP and bP. because when s/he has no access to a and b, s/he faces the Divisible computational Diffie-Hellman (DCDH) problem which is believed to have no polynomial time algorithm to compute. Hence, the known-key security property is achieved in our protocol.

- *Forward secrecy*: Compromising the long-term private keys of Alice and Bob will not reveal previously established session keys. It is obvious that the adversary $\mathcal{A}$ cannot compute T_A and T_B without knowing of R_A and R_B even with providing the long-term private keys of Alice and Bob. So, our protocol has perfect forward secrecy.

- *Key-compromise impersonation*: Suppose that an adversary $\mathcal{A}_{\mathcal{I}}$ has replaced Bob's public key with $P_B = x_e P$, where $x_e \in \mathbb{Z}_n^*$ is selected by himself, he could not compute the T_B or T_A without knowing of ephemeral short private keys a, b. Then, considering type II adversary, $\mathcal{A}_{\mathcal{II}}$ has known the KGC's master key s and Bob's partial key D_B, but he cannot generate $K_{AB}^1, K_{BA}^1, K_{AB}^2$ or K_{BA}^2 without knowing the values of ephemeral short private keys a, b and long-term private keys x_A and x_B of Alice and Bob, since he also cannot solve the (DCDH) problem.

- *Unknown key-share resilience*: Suppose an adversary $\mathcal{A}$ attempts to coerce Alice to share a session key with him, while Alice believes a session key is shared with Bob. For $\mathcal{A}$ to launch this attack successfully, he should force Alice and Bob to share the same secret. However, our protocol including the identities information of participating peers in computing the session key can prevent UKSR attack.

5.3 Efficiency Comparison

In this section, the comparison of our model against Ruffing *et al.* [19] is presented, the computational costs and communication overheads are highlighted in Table 2. For convenience, we define the following notations: T_H(the time complexity of one-way hash function); T_e(the time complexity of pairing operation); T_{mul}(the time complexity of a scalar multiplication operation of point).

As indicated in Table 2, Ruffing *et al.* [19] model requires two times bilinear pairing operation in session key agreement. However, a bilinear pairing operation is more time-consuming than other operations [7]. Its relative computation cost is approximately twenty times higher than that of the scalar multiplication over elliptic curve group [11]. Furthermore, the key escrow problem is addressed in our model. In other words, in our model the KGC cannot impersonate the user without being detected, while this feature is lacking in Ruffing *et al.* [19].

6 Experimental Results

We have conducted a series of repeated experiments using 512 × 512 24-bits standard RGB images: "Lena", "Baboon", "Peppers", "Jet" and "Barbara". The embedding capacity is measured in terms of bits. Steganographic method of this paper is implemented using Java 8 in environment as follows: HP-Compaq 610 laptop computer with Intel ® Core(TM)2 Duo CPU T5870 2.00GHz (2CPUs), 2.00 GHz, Memory RAM 1024MB, running on Windows 7 32-bit operating system. For evaluation test, we use Matlab R2013a 8.1.

The measurement tools used in this evaluation include Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). MSE and PSNR are mostly used for evaluating the robustness of steganographic system. The mean-squared error (MSE) between two images $A = \{a_1..a_M\}$ and $A' = \{a'_1..a'_M\}$ is given by Equation (2), where M is the number of pixels.

$$MES(A, A') = 1/M \sum_{i}^{m} (a_i - a'_i)^2 \tag{2}$$

PSNR is the ratio between the original signal and the stego signal in the image given in decibels. Formula (3) shows the PSNR test. For images $A = \{a_1..a_M\}$ and $A' = \{a'_1..a'_M\}$, and MAX equal to the maximum possible pixel value ($2^8 - 1 = 255$ for 8-bit image).

$$PSNR = 10 \log_{10}(\frac{MAX^2}{MSE(A, A')}) \tag{3}$$

As seen in Equations 2 and 3, there is an inverse relationship between MSE and PSNR, a low value of MSE give rise to higher value of PSNR, which signifies that a higher value of PSNR shows the higher quality of the image.

As indicated in Table 3, the average of PSNR values is 53.66 db, while MSE average is 0.48. This confirms that the proposed steganographic method is good in terms of invisibility of the embedded data. In other words, it is

Table 3: Experimental results of original and stego image

Image	Hiding capacity (bits)	PSNR	MSE
Lena	3,547,174	55.67	0.18
baboon	2,822,323	56.39	0.15
peppers	4,272,027	55.03	0.20
Jet	2,336,136	45.86	1.69
Barbara	3,909,601	55.34	0.19
Avg.	3,377,452	53.66	0.48

Table 4: Experimental results in the presence of Gaussian filter

Image	σ	MSE	PSNR
Baboon	0.1	0.1492602	56.391363
	0.2	0.1492602	56.391363
	0.25	0.1492602	56.391363
	0.27	0.1493619	56.388404

hard to distinguish between covertext and stegotext. Also graphical comparisons are presented in Figures 7, 8 and 9 for original and stego images and Figures 10, 11 and 12 for original and stego histograms.

In the following, the results of evaluating the robustness of proposed steganographic method against various kinds of sophisticated attacks are presented.

6.1 Gaussian Filtering

Robustness of proposed steganographic method is evaluated against Gaussian filtering attack with window size of 5×5. Table 4 presents the mean squared error (MSE) and peak signal-to-noise ratio (PSNR) in the presence of Gaussian filter with window size of 5×5 and variance (sigma σ) between 0.1 and 0.3 for Baboon stego image. As indicated in Table 4, the Gaussian filtering attack does not affect the robustness of hidden text h_{txt} by a considerable amount. The proposed steganographic method does not fail under Gaussian filter with window size 5×5 and variance $\sigma \leq 0.27$.

6.2 Bilinear Interpolation Image Rescaling

The results of rescaling raw image data using bilinear interpolation of proposed steganographic method are presented in Table 5. We have shown that the hidden text h_{txt} have not been affected by rescaling the stego image. Figure 13 shows the result of this operation using two different compression ratios.

Figure 7: Original and stego Lena from left to right respectively

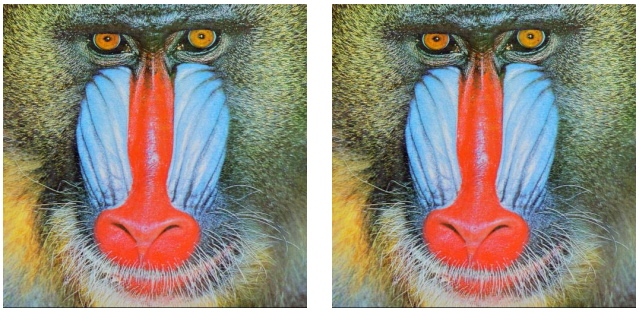


Figure 8: Original and stego Baboon from left to right respectively

Figure 9: Original and stego Peppers from left to right respectively

Table 5: Experimental results in the presence of Gaussian filter

Image	Compression ratio	MSE	PSNR
Lena	66%	0.176223	55.670159
	150%	0.176223	55.670159

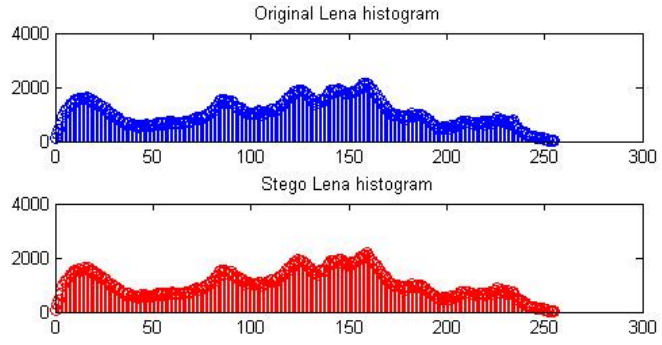


Figure 10: Original and stego Lena histograms

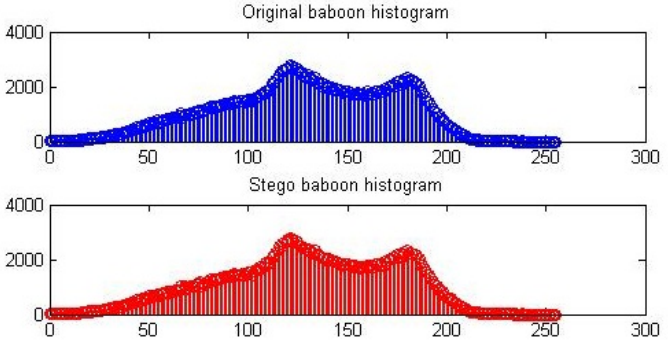


Figure 11: Original and stego Baboon histograms

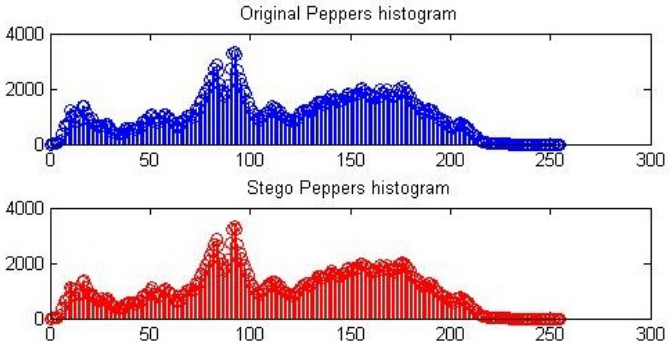


Figure 12: Original and stego peppers histograms

Figure 13: Rescaled image, bilinear interpolation, 66% and 150%' from left to right respectively

6.3 Pepper-Salt Noise Attack

Pepper-Salt noise causes on and off pixels. The results of evaluation of proposed steganographic method in the presence of Pepper-Salt noise are presented in Table 6. We adopted noise density between 0.000001 and 0.000005. As shown in Table 6, our steganographic method does not fail under Pepper-Salt noise with noise density ≤ 0.000005.

6.4 Pixel Difference Histogram Analysis

The results of difference histogram analysis are shown in Figure 14. From the figure, we observe that there are more numbers of bins which are close to 0 as compared to bins which are away from 0. Furthermore, the step pattern is not shown in the figure. Hence, the proposed steganographic method is robust against histogram analysis attack.

Table 6: Experimental results in the presence of pepper-salt noise

Image	noise density	MSE	PSNR
Baboon	0.000001	0.190282	55.336822
	0.0000015	0.190282	55.336822
	0.000005	0.298773	53.377383

6.5 Chi-Square Analysis

Chi-Square (χ^2) is a statistical test commonly used to calculate the average LSB and construct a table of frequencies and Pair of Values. Figure 15 shows the results of chi-square analysis on Baboon stego image. As the graph obtained fulfills the required range. Therefore, the proposed scheme successfully sustains this attack.

6.6 RS Analysis

RS analysis is one of the most reliable steganalysis which performs statistical analysis of the pixels to successfully detect the hidden message in an image. Figure 16 shows the results of RS analysis. From figure, we show that the difference between the relative number of regular groups (Non-overlapping groups) and the relative numbers of singular groups (Overlapping groups)is very small. The rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ are satisfied for Baboon stego image. This confirms that the proposed steganographic method is secure against RS attack.

6.7 Requirements Analysis and Comparison

Considering the importance and necessity of the security requirements of steganographic system that have been discussed in Section 1.2, we outline in this section how these requirements can be achieved using our proposed scheme.

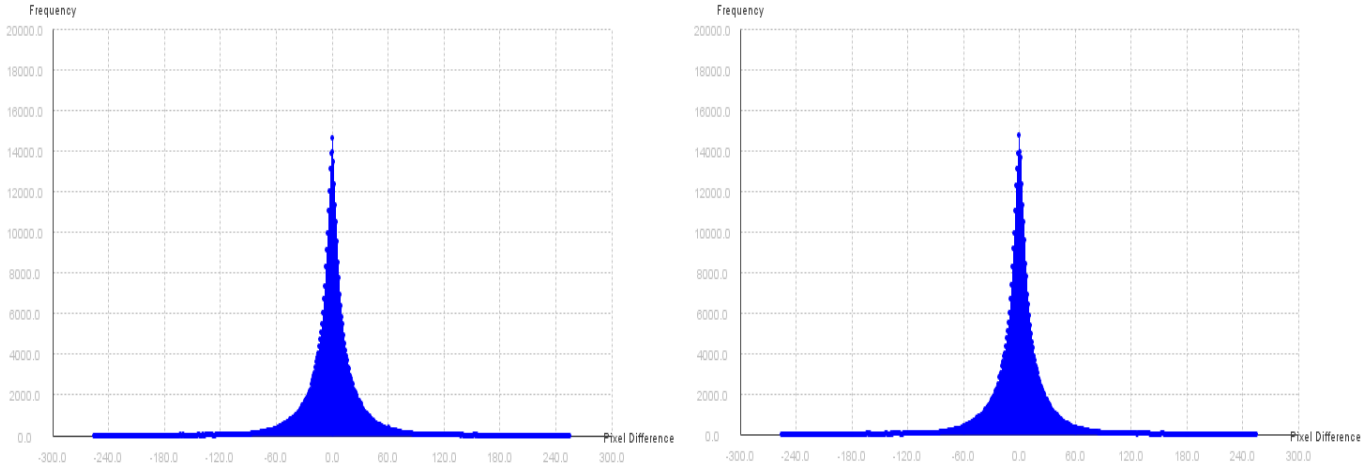


Figure 14: Pixel difference histogram analysis on baboon original and stego image from left to right respectively

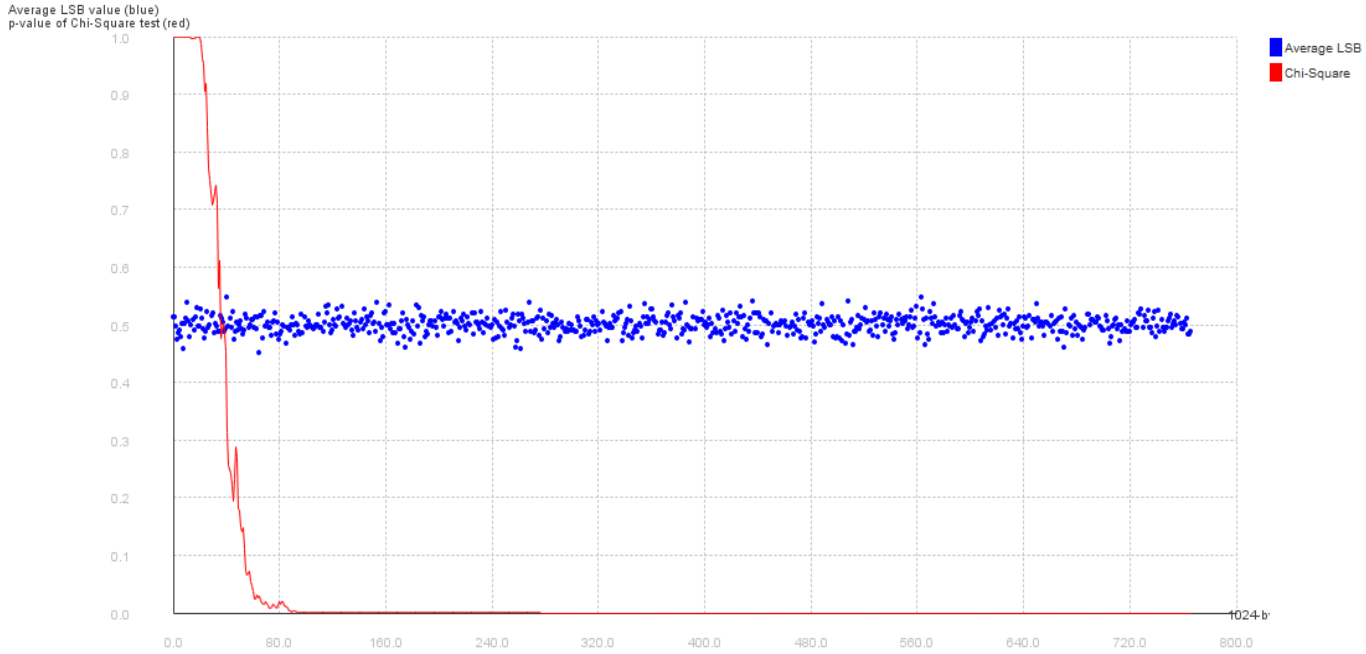


Figure 15: Chi-square attack from bottom to top on baboon stego image

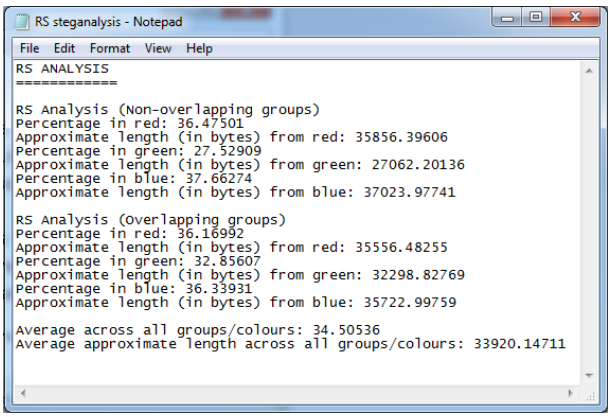


Figure 16: RS analysis on baboon stego image

- *Undetectability*: From the experimental results that have been carried out on all cover images, the PSNR values are greater than 70 db, while MSE values are very close to zero. Therefore, the undetectability of our steganographic method is achieved.
- *Robustness*: As proved in Sections 6.1, 6.2 and 6.3, the proposed steganographic method resists various kinds of sophisticated attacks.
- *Indistinguishability*: As mentioned in the analysis of undetectability, our steganographic method provides a high quality (PSNR) and a very small error rate (MSE) for all cover images compared with original images. Therefore, it is hard to distinguish between covertext and stegotext.
- *Security*: Assume that there is an attacker suspected in the stego-cover S_{txt}, then s/he performed statistical tests and discovered there is a hidden text into stego-cover S_{txt}, s/he cannot retrieve the hidden text h_{txt} because is stegoed using shared secret key. Thus, only legal user can destego the hidden text h_{txt}.

The comparison of hiding capacity and the obtained PSNR against [18, 24] are given in Table 7. From the table, the average hiding capacity and obtained PSNR of our steganographic method are more better than [18, 24].

Table 7: Comparison of hiding capacity achieved and the obtained PSNR

Cover image	Method	Hiding capacity (bits)	PSNR (db)
Lena	[18]	1,166,296	42.26
	[24]	2,045,260	42.40
	Our Method	3,547,174	55.67
Baboon	[18]	1,159,328	38.44
	[24]	1,956,789	38.25
	Our Method	2,822,323	56.39
Peppers	[18]	1,167,960	42.28
	[24]	2,110,148	41.99
	Our Method	4,272,027	55.03
Jet	[18]	1,165,184	42.60
	[24]	2,056,879	42.24
	Our Method	2,336,136	45.86
Avg.	[18]	1,164,692	41
	[24]	2,042,269	41
	Our Method	3,244,415	53

7 Conclusion

Public-key steganography allows two parties that have no prior knowledge of each other to communicate covertly over public channel. In this paper, we construct efficient certificateless public key steganography based on Distributed Discrete Wavelet Transform (DDWT) and Singular Value Decomposition (SVD). The experimental results show that the proposed steganographic method resists various kinds of sophisticated attacks. Furthermore, our scheme satisfies all stegosystem security requirements. Meanwhile it improves computational costs and communication overheads.

References

[1] S. Ahadpour, M. Majidpour, and Y. Sadra, "Public key steganography using discrete cross-coupled chaotic maps," *arXiv preprint arXiv: 1211.0086*, 2012.

[2] L. von Ahn, N. J. Hopper, "Public-key steganography," in *Advances in Cryptology (EUROCRYPT'04)*, LNCS 3027, pp. 323–341, Springer, 2004.

[3] S. S. Al-Riyami, and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the Cryptography (Asiacrypt'03)*, LNCS 2894, pp. 452–473, Springer-Verlag, 2003.

[4] R. J. Anderson and F. A.P. Petitcolas, "On the limits of steganography," *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, pp. 474-481, May 1998.

[5] M. Backes and C. Cachin, "Public-key steganography with active attacks," in *Proceedings of the Second International Conference on Theory of Cryptography (TCC'05)*, pp. 210–226, 2005.

[6] J. Baek, R. Safavi-Naini, W. Susilo, "Certificateless public key encryption without pairing," *Information Security*, pp. 134–148, 2005.

[7] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[8] R. P. S. Chauhan, R. Dwivedi, S. Negi, "Comparative evaluation of DWT and DT-CWT for image fusion

and de-noising," *International Journal of Applied Information Systems*, vol. 4, no. 2, pp. 40–45, 2012.

[9] P. Guillon, T. Furon, P. Duhamel, "Applied public-key steganography," in *Proceedings of Electronic Imaging*, pp. 38–49, 2002.

[10] D. He, J. Chen, J. Hu , "A pairing-free certificateless authenticated key agreement protocol," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 221–230, 2012.

[11] D. He, J. Chen, R. Zhang, "An efficient identity-based blind signature scheme without bilinear pairings," *Computers & Electrical Engineering*, vol. 37, no. 4, pp. 444–450, July 2011.

[12] N. J. Hopper, *Toward a Theory of Steganography*, Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2004.

[13] S. Katzenbeisser, F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Inc., Norwood, MA, USA., 2000.

[14] T. V. Le, "Efficient provably secure public keysteganography," *IACR Cryptology ePrint Archive*, article 156, 2003.

[15] T. V. Le, K. Kurosawa, *Efficient Public Key Steganography Secure Against Adaptively Chosen Stegotext Attacks*, Technical Report, Florida State University, 2003.

[16] T. V. Le, K. Kurosawa, "Bandwidth optimal steganography secure against adaptive chosen stegotext attacks," in *8th International Workshop on Information Hiding (IH'06)*, LNCS 4437, pp. 297-313, Springer-Verlag, 2007.

[17] C. H. Lin, J. S. Jen, and L. C. Kuo, "Distributed discrete wavelet transformation for copyright protection," in *The 7th International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 53–56, 2006.

[18] J. K. Mandal and D. Das, "Colour image steganography based on pixel value differencing in spatial domain," *International Journal of Information Sciences and Techniques*, vol. 2, no. 4, pp. 83–93, July 2012.

[19] T. Ruffing, J. Schneider and A. Kate, "Identity-based steganography and its applications to censorship resistance," *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 1461–1464, 2013.

[20] S. A. Seyyedi, V. Sadau and N. Ivanov, "A secure steganography method based on integer lifting wavelet transform," *International Journal of Network Security*, vol. 18, no. 1, pp. 124–132, Jan. 2016.

[21] G. J. Simmons, "The prisoner's problem and the subliminal channel," in *Advances in Cryptology (Crypto'83)*, pp. 51-70, 1984.

[22] R. S. Stankovic, B. J. Falkowski, "The Haar wavelet transform: its status and achievements," *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25–44, 2003.

[23] V. Strumpen, H. Hoffmann, A. Agarwal, *A Stream Algorithm for the SVD*, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, 2003.

[24] P. S. Vitthal , B. S. Rajkumar , P. R. Archana, "A novel security scheme for secret data using cryptography and steganography," *International Journal of Computer Network and Information Security*, vol. 2, pp. 36–42, 2012.

[25] A. Wahaballa, O. Wahballa, F. Li, M. Ramadan, Z. Qin, "Multiple layered securities using steganography and cryptography," *International Journal of Computers and Applications*, vol. 36, no. 3, 2014.

[26] S. Wang , Z. Cao, H. Bao, "Efficient certificateless authentication and key agreement (CL-AK) for grid computing," *International Journal of Network Security*, vol. 7, no. 3, pp. 342-347, Nov. 2008.

Osman Wahballa received the B.S. degree in electrical engineering and computer engineering from Karary University, Department of Electrical Engineering in 2006, Khartoum, Sudan, and the M.S. degree in M.Sc. in Computer Engineering, Information Security form University of Electronic Science and Technology of China in 2013, Chengdu, China. He is currently working toward the Ph.D. degree in computer science from University of Electronic Science and Technology of China. His current research interests include information hiding, steganography, and cryptography.

Abubaker Wahaballa is currently working as a Post-doctoral Fellow at School of Information and Software Engineering, University of Electronic Science and Technology of China UESTC. He received his PhD degree from UESTC in 2015. His current research interests include information security, cryptography, steganography, and DevOps.

Fagen Li Fagen Li received his Ph.D. degree in cryptography from Xidian University, Xian, China in 2007. He is now an associate professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His recent research interests include cryptography and network security. He has published more than 70 papers in the international journals and conferences.

Chunxiang Xu received her B.Sc., M.Sc. and Ph.D. degrees at Xidian University, in 1985, 1988 and 2004 respectively, P.R. China. She is presently engaged in information security, cloud computing security and cryptography as a professor at University of Electronic Science Technology of China (UESTC).

A Survey of Attribute-based Access Control with User Revocation in Cloud Data Storage

Chi-Wei Liu[1], Wei-Fu Hsien[2], Chou-Chen Yang[2], and Min-Shiang Hwang[1,3]
(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 Medical Research, China Medical University Hospital, China Medical University[3]
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan (R.O.C.)
(Received June 25, 2015; revised and accepted Aug. 27 & Sept. 23, 2015)

Abstract

Cloud storage service is one of cloud services where cloud service provider can provide storage space to customers. Because cloud storage service has many advantages which include convenience, high computation and capacity, it attracts the user to outsource data in the cloud. However, the user outsources data directly in cloud storage service that is unsafe when outsourcing data is sensitive for the user. Therefore, ciphertext-policy attribute-based encryption is a promising cryptographic solution in cloud environment, which can be drawn up for access control by the data owner to define access policy. Unfortunately, an outsourced architecture applied with the attribute-based encryption introduces many challenges in which one of the challenges is revocation. The issue is a threat to data security in the data owner. In this paper, we survey related studies in cloud data storage with revocation and define their requirements. Then we explain and analyze four representative approaches. Finally, we provide some topics for future research

Keywords: Access control, ciphertext-policy attribute-based encryption, cloud data storage, user revocation

1 Introduction

Cloud computing is a computing technology, and the internet has grown in recent years. It can share the software and hardware resource, and provides resources to a user's computer or mobile device. The user can obtain a more efficient service because cloud computing can integrate resources. Thus, cloud service providers have joined to build cloud environments and provide services to the user. Cloud service providers offer three services including Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The cost for users to rent cloud service is cheaper than the cost for users to build cloud environment [1].

Cloud storage service is the most common and popular service among many cloud services (e.g. Google Drive, Dropbox, Amazon S3 and Microsoft OneDrive) for general users. The user can pay to the cloud server provider based on the amount of usage. Then because cloud storage service provides to access cloud services from web service or applications that utilize the application programming interface (API) by mobile devices (e.g. laptop, table computer, and smart phones), it is convenient to use by users so to achieve ubiquitous service.

Although cloud storage service has many advantages, it also brings a lot of challenging issues which include efficacy and security [12, 17, 36, 44]. One of the serious challenges is protecting the confidentiality of the data. Because the traditional method means the user directly stores the data where data is not encrypted in the cloud storage server, the cloud storage server can understand the upload data of the user. Therefore, if these data are sensitive to users, this is unsafe. In order to ensure that it is safe for the user to upload data to the cloud storage server, a user utilizes an encryption method for processing sensitive data before the user uploads data to the cloud storage server.

For example, a user can utilize a symmetric-key algorithm to encrypt data before uploading to the cloud storage server. When the user needs data, he/she can download data and decrypt it by using a symmetric key. However, it is not suitable when the user shares data with the other users. Because the data owner needs to share their symmetric key with the shared user, the shared user can obtain data on the data owner's permission which contains all the data owner's data in the cloud storage server. Thus, this method is not secure in the situation.

A user chooses the other asymmetric-key algorithm

while the data owner uses a public key to encrypt data before uploading it to the cloud storage server. When the data owner needs data, he/she can download data and decrypt it by using a private key. In the shared situation, the data owner first downloads the shared data and decrypts it. Then, the data owner uses the shared user's public key to encrypt the shared data before uploading to the cloud storage server. However, this method has three problems: first the data owner needs to obtain the correct user's public key where the data owner can encrypt data. Second the same data stored will be repeated in the cloud storage server. Because the data owner will share the encrypted data to each user by their own public key, it will repeat the same data stored in the cloud storage server. Third, when the data owner shares data with a lot users, the data owner takes a lot of resources of the computation in the download and re-encrypted data. To solve these problems of how to design a method that can be able to get the correct user's public key, it only need to store one copy of the shared data in the cloud storage server, which reduces the data owner's resource of the computation.

In order to improve these problems, Sahai and Waters [33] proposed an attribute-based encryption (ABE) scheme where the scheme utilized a user's identity as attribute, and a set of attributes are used to encrypt and decrypt data. Their ABE scheme can resolve these problems including utilizing the attribute of a user's identity to make sure the user's public key, utilizing the ABE to reduce the duplication of data in the cloud storage service and the data owner only need to modify access policy where the data owner can reduce computing resources including downloading, decrypting, re-encrypting and reuploading the entire data.

In the access policy, the ABE has two categories: the key-policy attribute-based encryption (KP-ABE) [9] and the ciphertext-policy attributed encryption (CP-ABE) [4]. The KP-ABE scheme implies that the access policy is attached to the user's private key and use the user's set of attributes to describe the encrypted data. If a set of attributes of the privacy key satisfy the access policy, the user will decrypt the encrypted data. Otherwise, the user cannot obtain the encrypted data. The CP-ABE scheme implies that the access policy is associated with the encrypted data, and use the user's set of attributes to describe the user's private key. If a set of attributes of the encrypted data satisfies the access policy, the user will decrypt the encrypted data. Otherwise, the user cannot obtain the encrypted data.

Nowadays, the outsourced data needs flexible access control for users. The traditional method of access control is a trusted cloud server responsible for the definition and implementation of access control policies. However, users want to be able to share sensitive data and define access policies and the implementation of his/her data with a group of people of their choice. Therefore, it is a desirable method that the access policy of the data will be defined by the data owner.

CP-ABE provides a scalable method of encrypting data where the encrypted user defines the attribute set, and then the decrypted user needs to hold the attribute set to decrypt the ciphertext [4]. Therefore, different users are allowed to decrypt different data block in the different access policies. This effectively reduces depending on the cloud storage server for preventing unauthorized data access.

There are many extended ABE related researches including multi-authority, accountability, proxy re-encryption and revocation. In each ABE scheme, the user needs to get a secret key from the trusted authority which can prove his/her identity, and use the secret key to decrypt data. However, because the authority can decrypt all ciphertexts in a single-authority ABE scheme, the user utilizes a single-authority ABE scheme, which is not proper in the situation there are different departments. Therefore, Chase [7] proposed the first multi-authority ABE scheme which extends a single-authority ABE scheme. Then, the multi-authority ABE schemes were proposed in [6, 7, 8, 11, 18, 26, 27]. In order to achieve secure access control, the ABE scheme needs to prevent accountable key abuse which includes an illegal key sharing among colluding users and misbehavior of the semi-trusted authority containing illegal key distribution or re-distribution. Accountable ABE can be divided in two categories including accountable KP-ABE scheme [37, 42] and accountable CP-ABE scheme [20, 21, 22]. In order to make sharing more efficient, the proxy re-encryption (PRE) is proposed because the user can delegate other to re-encrypt data. However, when the user is not online, the ABE scheme cannot directly use the capability of decryption to others. Therefore, the attribute-based PRE (ABPRE) scheme is proposed [10, 23, 24, 28, 34] which combines the proxy re-encryption with the ABE. A user is able to delegate designated users to decrypt the re-encrypted ciphertext by the associated attributes of designated users.

There are mainly two ways to realize revocation: one is the indirect revocation method [4, 5, 14, 15, 31, 38, 43], and the other is the direct revocation method [2, 25, 30].

Indirect revocation method means the data owner delegates authority to execute revocation which releases a key update material periodically in such a way that only non-revoked users can update their keys. An advantage of the indirect revocation method is that the data owner does not need to know the revocation list. However, the disadvantage of the indirect revocation method is that it requires communication from the authority to all non-revoked users at all time slots in the key update phase. Some related attribute revocable ABE schemes [4, 5, 14, 15, 31, 38, 43] which used the indirect method have been proposed. The direct revocation method means the data owner executes direct revocation which specifies the revocation list while encrypting the ciphertext. An advantage of the direct revocation method over the indirect revocation one is that it does not include the key update phase for all non-revoked users interact-

ing with the authority. However, the disadvantage of the direct revocation method is that it needs the data owner to manage the current revocation list because it is a troublesome problem. Some related attribute revocable ABE schemes [2, 25, 30] which used the direct method have been proposed. Attrapadug and Imai [3] first proposed a hybrid ABE (HR-ABE) scheme which utilized the advantage of both indirect and direct methods. Their scheme allows the data owner to select the encrypted scheme including indirect or direct method. Then, their scheme supports user revocation, but it is unable to achieve attribute revocation. However, it increases the user's secret key in length.

Although there are many ABE related studies, we will focus on user revocation mechanism in the cloud data storage. However, this introduces a number of challenges which utilized ABE to solve the outsourced data. One of the challenge is the revocation of attribute and user. The revocation issue is more difficult in ABE system because each attribute is shared by multiple users. When the revocation of any attribute and single user is in an attribute group, it would affect the other users in the group. It will generate a bottleneck for the rekeying procedure and secure threat in ABE system. Therefore, in this paper, we will survey the problem in attribute-based data access control using CP-ABE for a data outsourcing system.

1.1 Requirement

According to these studies, they provide the basic requirements of function and performance. In our paper, we classify and describe these requirements. Then we use these requirements to analyze the existing scheme in Section 4.

Functional evaluation

1) Data confidentiality: The data owner encrypts the data before uploading data to the cloud. Therefore, the unauthorized user and cloud storage server cannot know the encryption data.
2) Fine-grained access control: Each user respectively has own access right which may be different for each user. Even if the users exist in the same group, their access right may not be the same.
3) Scalability: When the authorized users increase, the cloud storage server can execute efficiently. Therefore, the number of authorized users cannot affect the performance of the cloud storage server.
4) User revocation: If the user leaves the group, the scheme can revoke the user's access right from the cloud storage server. The revoked user cannot access any shared data in the group, because the user does not have access right.
5) Collusion resistant: The revoked user cannot collude with the cloud storage server to obtain the encrypted data which the data owner before sharing the data with the revoked user.
6) Forward secrecy: In an attribute which satisfies the access policy, any user drops the attribute which can be prevented from accessing the plaintext of the subsequent data exchanged after the user drops the attribute.
7) Backward secrecy: In an attribute which satisfies the access policy, any user holds the attribute which can be prevented from accessing the plaintext of the previous data exchanged before he holds the attribute.

Performance evaluation

1) Computing cost: In order to achieve an efficient public auditing, we will analyze the client, TPA and cloud storage service cost on the computing resources.
2) Storage cost: Because the client will upload data to the cloud storage service without the local copy of data files, we will analyze the client, TPA and cloud storage service cost on the storage spaces.

1.2 Our Contribution

Our contribution can be summarized as the following three aspects: First, we survey the previous researches of attribute-based access control with user revocation in the cloud. Then our paper collects and explains basic requirements in the mechanism. Second, we propose four representative approaches and analyze these approaches by our collected requirements. Third, we summarize the conclusion from the analysis and propose research direction in future work.

1.3 Organization

The rest of paper is organized as follows: In Section 2, we review the related work of revocation. We discuss the representative approaches of user revocation in detail in Section 3. In Section 4, we analyze the basic requirement in the representative approaches. Finally, we summarize and discuss the future work in Section 5.

2 Related Work

Recently, some attribute revocable ABE schemes have been proposed [4, 5, 31]. Piretti et al. [31] proposed the time rekeying mechanism where each attribute is associated with an expiration time. Bethencourt et al. [4] improved this solution which utilized the user secret key with a single expiration time. The users need to update their keys frequently, and the authority has lower resource of computation. Boldyreva et al. [5] proposed an efficient revocation scheme for IBE, which utilized binary tree to

build data structure. Their scheme did not use CP-ABE scheme. These schemes were named a coarse-grained revocation because these scheme are not able to immediately rekey on any member change. Furthermore, these schemes have two main problems on scalability and security degradation which include forward and backward secrecy [13, 14, 32, 38]. In the scalability problem, the key authority periodically distributes an update information of key to update the non-revoked users' keys. However, the previous revocation did not consider the scalable distribution where the updated attribute keys distributes the group of users who share the attributes.

In the ABE system, an attribute is supposed to be shared by a group of users. Then it is a considerable situation where the members may change frequently in the group. However, a new user might be able to access the previous encrypted data before the user comes to hold the attributes until the data is re-encrypted with the update attribute keys by periodic rekeying which was named backward secrecy. On the other hand, a revoked user would be able to access the encrypted data until the next expiration time which was named forward secrecy. Therefore, the uncontrolled period has serious vulnerability.

Then many CP-ABE schemes [13, 14, 38, 40, 43] with immediate attribute revocation have been proposed instead of periodic or timed revocation. Yu et al. [43] proposed a scheme which utilized proxy re-encryption with CP-ABE. However, their scheme needed to spend the revocation cost highly where the system public key and users' secret key changed. Hur and Xie et al. [13, 14, 38] proposed efficient attribute revocation schemes which utilized the tree of access policy to encrypt data. The cloud storage server needs to spend high resource of computation because the cloud storage server re-encrypt all the ciphertext with a new generated encrypting key during the attribute revocation. Yang et al. [40] proposed an attribute revocation scheme in CP-ABE where the authority updated the ciphertext and produce new keys that include the new version key, update key, and secret key. However, the authority needed to spend high resource of computation in their scheme.

Yang et al. [41] proposed an improved scheme in CP-ABE which extended multi-authority with attribute revocation. However, the authority needed to enhance efficiency.

User revocation has been observed in the many practical ABE system. Because users may change their attributes frequently, the mechanism of user revocation is essential in many group-based applications [29, 32]. Ibraimi et al. [15] proposed a fine-grained user-level revocation scheme which utilized negative clauses in ABE scheme. When a user is revoked, the user is added to AND of negation of revoked user identities. However, their scheme lacks efficiency of the implementation.

Golle et al. [35] proposed a user revocable scheme which utilized KP-ABE scheme. However, their scheme only supports that the number of attributes associated with a ciphertext is exactly half of the universe size. The previous user-revocable schemes have a drawback on the availability. The availability means the granularity of the user access control between attribute-level or system-level revocation. If a user is revoked from a single attribute group, the user would lose all the access rights on the data sharing system. The previous schemes [25, 30] executed user revocation on system-level, which means the user was revoked from the whole system. However, the system-level revocation scheme is not suitable because the revoked user still has the access right of other data in the system. Therefore, the attribute-level user access control will suit in many practical data outsourcing situation. Attrapadung et al. [2] and Junod et al. [16] proposed user revocation ABE scheme which utilized broadcast encryption scheme on ABE scheme. In Attrapadung et al.'s scheme, the data owner should take full charge of maintaining all the membership lists for each attribute group. However, it is not applicable to the cloud storage architecture because the data owner will no longer be directly in control of data. In the Junod et al.'s scheme, user revocation is achieved by updating the set of identity attributes. However, every user has an identity attribute except for system attribute, which causes the ciphertext growing linearly with the users. Xu et al. [39] proposed a scheme of dynamic user revocation which utilized a delegation key for the cloud storage server to re-encrypt ciphertext. However, the cloud storage server has full control of a revocation list for revoked users. When a user is assigned the revocation list, he/she will lose all access right to the data. Li et al. [19] proposed a revocable identity-based encryption (IBE) scheme in the outsourcing computation. The cloud storage server executes updating secret keys for non-revoked users on the revocation phase. However, two factors that include identities of revoked users and time periods are needed by the secret key generator and the cloud storage server. Zu et al. [45] proposed a new CP-ABE scheme which utilized the access structure of linear secret sharing scheme (LSSS) to define access control in cloud storage service.

3 Representative Approaches

Before introducing representative approaches, we explain the system model in Figure 1, and list all notations (as shown in Table 1) used in this paper.

- The Data Owner: An individual consumer or organization has a lot of data files and needs to store in the cloud. The data owner is responsible for defining (attribute-based) access policy, and encrypting own data under the policy before distributing it.

- Users: A user want to access the data from the data owner. If a user possesses a set of attributes satisfying the access policy of the encrypted data, he/she will be able to decrypt the ciphertext and obtain the data.

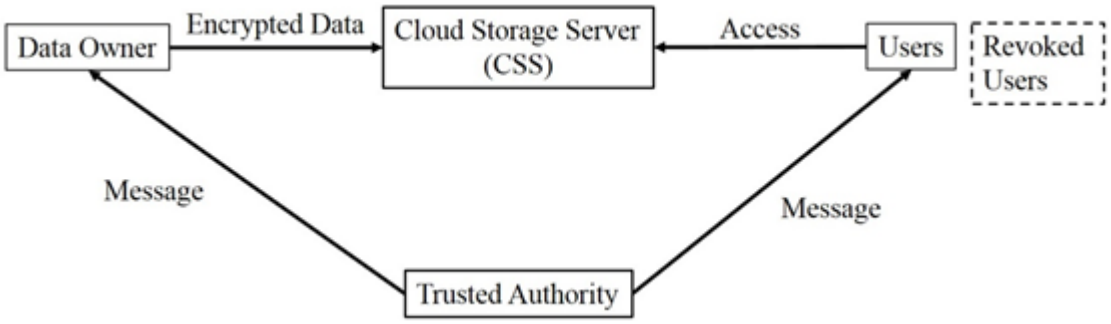


Figure 1: The cloud storage architecture of attribute-based access control

- Cloud Storage Server (CSS): A cloud service provider has huge storage space, computation resource and shared service to provide the clients. It is responsible for controlling the data storage in outside users' access, and provides the corresponding contents.
- Trusted Authority (TA): A trusted organization has expertise and capabilities that the clients do not have. It generates public and secret parameters for ABE, then responsible for issuing, revoking and updated attribute key for the user. It also grants differential access rights to individual users based on their attributes.

3.1 Hur and Noh's Scheme

Hur and Noh [14] was the first to propose an efficient revocation scheme which utilized the access structure of tree to define access control in data outsourcing. Their scheme used CP-ABE scheme to encrypt data and define access policy by the user which was flexible in sharing data with other users. In order to achieve the fine-grained access control, they utilized dual encryption mechanism which took advantage of the ABE and selective group key distribution in each attribute group. They considered the previous user revocable schemes have a limitation which means the granularity of the user access control between attribute-level or system-level revocation. If a user is revoked from a single attribute group in the previous studies [25, 30], the user would lose all the access rights on the system, which means system-level revocation. In the attribute-level revocation, if a user is revoked from a single attribute group, the user would only lose the access right of the attribute group. They proposed the fine-grained revocation to improve coarse-grained revocation because the coarse-grained revocation cannot immediately rekey on any member. The fine-grained revocation can avoid forward and backward secrecy.

Next we will describe their scheme including setup, key generation, data encryption, data re-encryption, data decryption and key update phase.

Setup Phase

The authority chooses two random values α, $\beta \in Z_p^*$, and generates the public parameter $PP = (G_1, g, H_1, h = g^\beta, e(g,g)^\alpha)$ and the master key $MK = (\beta, g^\alpha)$.

Key Generation Phase

The authority uses the master key MK, a set of attributes $\wedge$ and a set of user indices U to generate an attribute key for each user. It chooses a random value $r \in Z_p^*$ which is unique to each user, and a random value $r_j \in Z_p^*$ for each attribute $\lambda_j \in \wedge$ to compute the user u_t's private key $SK_t = (D = g^{(\alpha+r)/\beta}, \forall \lambda_j \in \wedge\text{: } D_j = g^r \cdot H_1(\lambda_j)^{r_j}, D'_j = g^{r_j})$.

The authority generates attribute keys for a set of users U. It uses a set of attributes $\wedge$ and a set of user indices U to generate an attribute keys for each user that identifies with that set $\wedge$. It sends the attribute group AG_j for each attribute $\lambda_j \in \wedge$ to the CSS. For example, if users u_1, u_2, u_3 are connected with $\{\lambda_1, \lambda_2, \lambda_3\}, \{\lambda_2, \lambda_3\}, \{\lambda_1, \lambda_3\}$, respectively. Then the authority sends $AG_1 = \{u_1, u_3\}, AG_2 = \{u_1, u_2\}, AG_3 = \{u_1, u_2, u_3\}$ to the CSS.

The CSS generates key encrypting keys (KEKs) for users in U. It constructs a binary KEK tree for the universe of users μ which will be used to distribute the attribute group keys to users in $U \in \mu$. It assigns each user u_i to the leaf node of the KEK tree, and generates random keys for each leaf node and internal node. Therefore, each user $u_t \in U$ receives the path keys $PathKey_t$ where the path is from the leaf node to the root node.

In the KEK tree (see Figure 2), each node ν_j of the tree holds as KEK, denoted by KEK_j. A set of KEK_j on the path nodes from a leaf to the root are named path keys. For example, the user u_3 stores the path key $PathKey_3 = \{KEK_{10}, KEK_5, KEK_2, KEK_1\}$. Each user u_i is assigned to the leaf node of the KEK tree. Random keys are generated and assigned to each leaf node and internal node.

Each user $u_t \in U$ receives the path keys $PathKey_t$ from its leaf node to the root node of the tree se-

Table 1: Notations

Notation	Significance
G_1, G_2, G_T	A multiplicative cyclic group
e	A bilinear map $e : G_1 \times G_2 \to G_T$
g	A generator of group G_1
p	The prime order of group G_1
q	A much smaller prime than p
H_1	A hash function $H_1 : \{0,1\}^* \to G_1$
H_2	A hash function $H_2 : \{0,1\}^* \to Z_p$
H_3	A hash function $H_3 : G_1 \to Z_p$
M	The message
m_i	A data block of the shared data and will be split into k elements
μ	The universe of users
L	The universe of descriptive attribute
G	The universe of such attribute groups
$\wedge$	A set of attribute
U	A set of user indices
AG	A set of attribute group
K_{λ_i}	The attribute group key

curely. The CSS uses the path keys KEKs to encrypt attribute group keys K_{λ_i} for each AG_i in the re-encryption phase.

Data Encryption Phase

The data owner wants to upload data M to the CSS and sharing data, he/she defines the tree access structure T over the universe of attributes L, and encrypts the data under T.

The data owner chooses a polynomial q_x where x is each node in the access tree T. These polynomials are chosen in a top-down mode which is from the root node R. In the access tree T, the degree d_x of the polynomial q_x be set one less than the threshold value k_x of the node as $d_x = k_x - 1$. Therefore, the root node R is chosen and a random value $s \in Z_p^*$ and set $q_R(0) = s$. Then the root node R sets d_R and other points of the polynomial q_R randomly to define q_R. Any other node x sets $q_x(0) = q_{parent(x)}(index(x))$ and randomly chooses d_x and other points to define q_x.

Then the data owner uses the public parameter PP and the tree of access structure to encrypt the message $M \in G_T$. Therefore, the ciphertext is $CT = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H_1(\lambda_y)^{q_y(0)})$ where Y be the set of leaf nodes in the access tree T.

Data Re-Encryption Phase

The CSS uses a set of the membership information for each attribute group $AG \in G$. The attribute group of the access tree is embedded in CT before distributing outsourced data CT. The re-encryption executes user access control from each attribute group on top of the outsourced ciphertext which was encrypted under the attribute-level access control policy by the data owner.

The CSS chooses a random value $K_{\lambda_y} \in Z_p^*$ in the attribute group $AG_y \in AG$ and re-encrypts CT. Therefore, the re-encrypted ciphertext is $CT' = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = (H_1(\lambda_y)^{q_y(0)})^{K_{\lambda_y}})$ where Y is the set of leaf nodes in the access tree T. Then the CSS selects the root nodes of the minimum cover sets in the KEK tree which can include all of the leaf nodes connected with users in $AG_i \in AG$. The $KEK(AG_i)$ is constructed from a set of KEKs which include the root nodes of subtrees AG_i. For example, if the attribute groups $AG_i = \{u_1, u_2, u_3, u_4, u_7, u_8\}$, the $KEK(AG_i) = \{KEK_2, KEK_7\}$ because ν_2 and ν_7 are the root node of the minimum cover sets which can cover all of the users in AG_i. If any user $u \notin AG_i$, they would not know any KEK in $KEK(AG_i)$.

Finally, the CSS generates a header message $Hdr = (\forall y \in Y : \{E_K(K_{\lambda_y})\}_{K \in KEK(AG_y)})$ where $E_K(M)$ is a symmetric encryption of a message M under a key K. This encryption is employed for the method to deliver the attribute group keys to valid users. The encryption is $E_K : \{0,1\}^k \to \{0,1\}^k$ a block cipher, where k is the length of the key K. Finally, when the CSS receives the data request from a user, the CSS sends (Hdr, CT') to the user.

Data Decryption Phase

When a user receives the ciphertext (Hdr, CT') from the CSS, he/she first obtains the attribute group keys for all attribute in $\wedge$ that the user holds from Hdr. If a user $u_t \in AG_j$ has a valid attribute λ_j, he/she

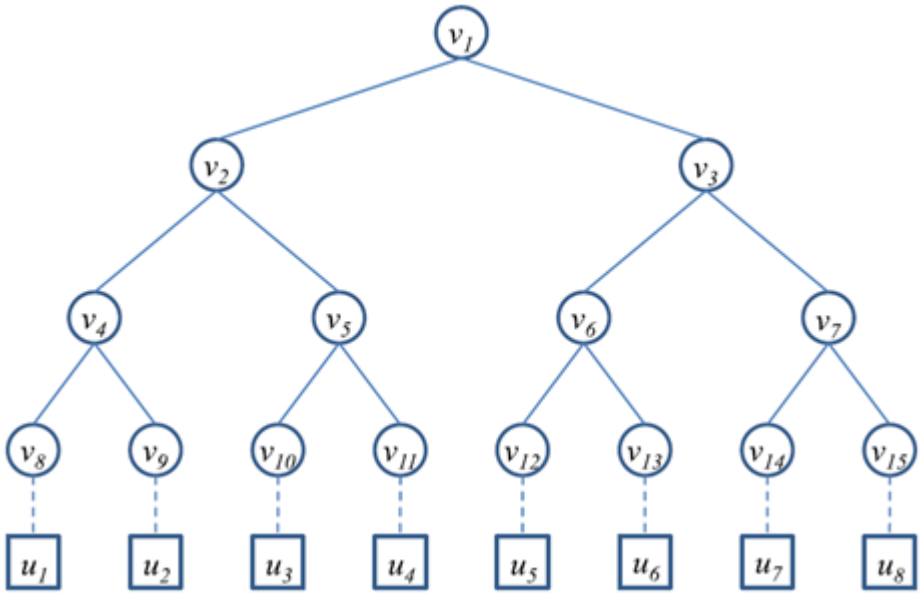


Figure 2: The KEK tree for attribute group key distribution

can decrypt the attribute group key K_{λ_j} from Hdr using a KEK that is common in $KEK(AG_j)$ and $PathKey_t$ where $KEK \in KEK(AG_j) \cap PathKey_t$.

For example, in the attribute groups $AG_i = \{u_1, u_2, u_3, u_4, u_7, u_8\}$, u_3 can decrypt the K_{λ_j} using the path key $KEK_2 \in PathKey_3$. Then the user u_t updates its secret key with the attribute group keys as follows:

$$\begin{aligned} SK_t &= (D, D_j, D'_j) \\ D &= g^{(\alpha+r)/\beta} \\ D_j &= g^r \cdot H_1(\lambda_j)^{r_j} \\ D'_j &= (g^{r_j})^{1/K_{\lambda_j}}, \forall \lambda_j \in \wedge. \end{aligned}$$

Then the user uses a private key SK and K_{λ_x} to decrypt the encrypted ciphertext CT' in the recursive function as $DecryptNode(CT', SK, x)$.

If the node x is a leaf node and $\lambda_x \in \wedge$ and $u_t \in AG_x$, then it computes

$$\begin{aligned} & DecryptNode(CT', SK, x) \\ = & \frac{e(D_x, C_x)}{e(D'_x, C'_x)} \\ = & \frac{e(g^r H(\lambda_x)^{r_x}, g^{q_x(0)})}{e((g^{r_x})^{1/K_{\lambda_x}}, (H(\lambda_x)^{q_x(0)})^{K_{\lambda_x}})} \\ = & e(g,g)^{rq_x(0)}. \end{aligned}$$

If $u_t \notin GA_x$, the user u_t cannot compute the values $e(g,g)^{rq_x(0)}$, as the exponent of D'_x in SK cannot include the inverse of the exponent K_{λ_x} of C'_x. If $\lambda_x \notin \wedge$ or $u_t \notin GA_x$, the $DecryptNode(CT', SK, x)$ will output invalid.

If the node x is a non-leaf node, the $DecryptNode(CT', SK, x)$ can be named from all nodes z which are children of x. For all nodes z call $DecryptNode(CT', SK, z)$ which use Lagrange coefficient to compute and obtain $e(g,g)^{rq_x(0)}$.

Therefore, if the access tree T is satisfied by $\wedge$, the user has valid memberships for each attribute group AG_i for all $\lambda_i \in \wedge$. Let $A = DecryptNode(CT', SK, R) = e(g,g)^{rs}$.

Finally, the user decrypts the ciphertext.

$$\begin{aligned} \frac{\tilde{C}}{e(C,D)/A} &= \frac{Me(g,g)^{\alpha s}}{e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}} \\ &= M. \end{aligned}$$

Key Update Phase

When a user comes to hold or drop an attribute, the corresponding key should be updated to avoid backward and forward secrecy on the previous or subsequent encrypted data. The key update procedure is executed by the authority when the user requests to join or leave on the attribute group. The authority receives the request, and sends the updated membership list of the attribute group to the CSS. Then the CSS receives the update request, and computes the corresponding attribute group key.

The CSS selects a random value $s' \in Z_p^*$ and a random value K'_{λ_i} which is different from the previous attribute group key $K_{\lambda_i} \neq K'_{\lambda_i}$. Then it re-encrypts the ciphertext CT using the public parameters PP

and public key PK as

$$\begin{aligned} CT' &= (T, \tilde{C}, C, C_i, C_i') \\ \tilde{C} &= Me(g,g)^{\alpha(s+s')} \\ C &= h^{s+s'} \\ C_i &= g^{q_i(0)+s'} \\ C_i' &= (H_1(\lambda_i)^{q_i(0)+s'})^{K'_{\lambda_i}} \end{aligned}$$

$\forall y \in Y : C_y = g^{q_i(0)+s'}, C_y' = (H_1(\lambda_y)^{q_y(0)+s'})^{K_{\lambda_y}})$.

In the other attribute groups, the attribute group keys do not necessarily need to be uploaded because they will not be affected by the membership changes.

The CSS chooses a new minimum set to cover the original attribute group AG_i, and the new set includes a new joining user who comes to hold an attribute λ_i (or exclude a leaving user who come to drop an attribute λ_i).

The CSS generates a new header message with the updated $KEK(AG_i)$ as

$$\begin{aligned} Hdr = (&\{E_K(K'_{\lambda_i})\}_{K \in KEK(AG_i)}, \\ &\{E_K(K_{\lambda_y})\}_{K \in KEK(AG_i)}, \forall y \in Y). \end{aligned}$$

Finally, the CSS responds new header message and the ciphertext.

3.2 Hur's Scheme

Hur [13] proposed an improved security scheme which considered a key escrow problem and user revocation in attribute-based data sharing. Their scheme used CP-ABE scheme to encrypt data and define access policy by the user which was flexible in sharing data with other users. Because the authority generates users' private keys by using the authority's the master key to users' associated set of attributes in the attribute-based encryption, the authority can decrypt every ciphertext addressed to specific users. This problem could generate a potential threat in the data sharing system of data confidentiality or privacy. Therefore, they designed the scheme where the authority and the CSS generated the user's secret key together that could avoid the key escrow problem. Then they considered the key revocation where the user may change their associate attributes. Therefore, the key revocation or update for each attribute is necessary to make system secure.

Next we will describe their scheme including setup, key generation, data encryption, data re-encryption, data decryption and key update phase.

Setup Phase

The authority chooses a random value $\beta \in Z_p^*$ and computes $h = g^\beta$. The public parameter $PP = (G_1, g, H_1, H_3)$, the public key $PK_A = h$ and the master key $MK_A = \beta$.

The CSS chooses a random value $\alpha \in Z_p^*$. The public key $PK_C = e(g,g)^\alpha$ and the master key $MK_C = g^\alpha$.

The CSS chooses another random value $\Upsilon \in Z_p^*$, and generates another public key $PK_C^{agree} = g^\Upsilon$ while keeping Υ as a secret.

Key Generation Phase

The authority needs to authenticate a user u_t which exists in U. If the result is true, the authority chooses a random value $r_t \in Z_p^*$ which is a unique secret for the user. Then the authority and the CSS construct a secure 2PC protocol, which combine the values (r_t, β) from the authority with the value α from the CSS. Therefore, the secure 2PC protocol is the value $x = (\alpha + r_t)\beta$.

1) The CSS chooses a random value $\tau \in Z_p^*$, computes $A = g^{\frac{x}{\tau}} = g^{\frac{(\alpha+r_t)\beta}{\tau}}$, and then sends it to the authority.
2) The authority computes $B = A^{1/\beta^2} = g^{\frac{\alpha+r_t}{\tau\beta}}$, and sends it to the CSS.
3) The CSS generates a personalized key component $D = B^\tau = g^{\frac{\alpha+r_t}{\beta}}$.
4) The authority uses a set of attributes $\wedge$ that a user u_t is entitled to have, and generates a set of attribute keys identified with that set and the secret value r_t. The authority chooses a random value $r_j \in Z_p^*$ for each attribute $\lambda \in \wedge$. Then it computes a user u_t's the attribute keys $SK_{A,u_t} = (\lambda_j \in \wedge : D_j = g^{r_t} H_1(\lambda_j)^{r_j}, D'_j = g^{r_j})$ to the CSS.
5) The CSS's personalized key component SK_{u_t} for a user u_t as $SK_{C,u_t} = D = g^{(\alpha+r_t)/\beta}$. Then the user u_t can obtain its whole secret key

$$\begin{aligned} SK_{u_t} &= (SK_{C,u_t}, SK_{A,u_t}) \\ &= (D, D_j, D'_j). \\ D &= g^{(\alpha+r_t)/\beta} \\ D_j &= g^{r_t} \cdot H(\lambda_j)^{r_j}, \forall \lambda_j \in \wedge \\ D'_j &= g^{r_j}. \end{aligned}$$

The CSS also generates another encrypting key (KEK) $SK_{u_t}^{agree} = H(ID_t)^\Upsilon = Q_t^\Upsilon$ for the user, which will be used for selective attribute group key distribution.

Data Encryption Phase

The data owner wants to upload data M to the CSS and sharing data, he/she defines the tree access structure T over the universe of attributes L, and encrypts the data under T.

The data owner chooses a polynomial q_x where x is each node in the access tree T. These polynomials are chosen in a top-down method which is from the root node R. In the access tree T, the degree d_x of the polynomial q_x is set one less than the threshold value k_x of the node as $d_x = k_x - 1$. Therefore, the root node R chooses a random value $s \in Z_p^*$ and

set $q_R(0) = s$. Then the root node R sets d_R other points of the polynomial q_R randomly to define q_R. Any other node x sets $q_x(0) = q_{parent}(x)(index(x))$ and randomly chooses d_x other points to define q_x.

The data owner uses the public parameter and the tree of access structure to encrypt the message $M \in G_T$. Therefore, the ciphertext is $CT = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H_1(\lambda_y)^{q_y(0)})$ where Y is the set of leaf nodes in the access tree T.

Data Re-encryption Phase

The CSS uses a set of the membership information for each attribute group $AG \subseteq Q$. The attribute group of the access tree is embedded in CT before distributing outsourced data CT. The re-encryption executes user access control from each attribute group on top of the outsourced ciphertext which was encrypted under the attribute-level access control policy by the data owner.

The CSS chooses a random value $K_{\lambda_y} \in Z_p^*$ in the attribute group $GA_y \in GA$ and re-encrypts CT. Therefore, the re-encrypted ciphertext is $CT' = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = (H_1(\lambda_y)^{q_y(0)})^{K_{\lambda_y}})$ where Y is the set of leaf nodes in the access tree T. Then, it selects $\rho, R \in Z_p^*$, and for all $u_t \in AG$ computes $x_t = H_3(e(Q_t^\rho, PK_C^{agree}))$. For all $AG_y \subset AG$ constructs the polynomial function $f^y(x) = \prod_{i=1}^{m}(x - x_i) = \sum_{i=0}^{m} a_i x^i (\mathrm{mod} p)$, where $AG_y = \{u_1, u_2, \cdots, u_m\}$ and the exponential function $\{P_0, \cdots, P_m\} \equiv \{g^{a_0}, \cdots, g^{a_m}\}$, where m is the number of users in the attribute group. It constructs $Hdr_y = \{K_{\lambda_y} \cdot P_0^R, P_1^R, \cdots, P_m^R\}$ and generates a header message $Hdr = (g^\rho, \cdots, \forall y \in Y : Hdr_y)$. Finally, when the CSS receives the data request from a user, the CSS sends (Hdr, CT') to the user.

Data Decryption Phase

A user receives the ciphertext (Hdr, CT') from the CSS, he/she first obtains the attribute group keys for all attributes in $\wedge$ that the user holds from Hdr. If a user $u_t \in AG_j$ has a valid attribute λ_j, he/she can decrypt the attribute group key K_{λ_j} from Hdr. The user u_t uses the KEK $SK_{u_t}^{agree}$ and g^ρ and computes $x_t = H_1(e(g^\rho, SK_{u_t}^{agree}))$. Then, the user u_t computes $K_{\lambda_j} \cdot P_0^R \cdot \prod_{i=1}^{m}(P_i^R)^{x_t^i} = K_{\lambda_j} \cdot g^{Rf^j(x_t)} = K_{\lambda_j}$, where $m = |AG_j|$. The user u_t updates its private key with the attribute group keys $SK_{u_t} = (D = g^{(\alpha + r_t)/\beta}, \forall \lambda_j \in \wedge : D_j = g^{r_t} \cdot H(\lambda_j)^{r_j}, D'_j = (g^{r_j})^{(1/K_{\lambda_j})})$.

Then, the user uses private key SK and K_{λ_x} to decrypt the encrypted ciphertext CT' in the recursive function as $DecryptNode(CT', SK, x)$.

If the node x is a leaf node and $\lambda_x \in \wedge$ and $u_t \in AG_x$, then it computes

$$\begin{aligned} & DecryptNode(CT', SK, x) \\ = \ & \frac{e(D_x, C_x)}{e(D'_x, C'_x)} \\ = \ & \frac{e(g^r \cdot H(\lambda_x)^{r_x}, g^{q_x(0)})}{e((g^{r_x})^{\frac{1}{K_{\lambda_x}}}, (H(\lambda_x)^{q_x(0)})^{K_{\lambda_x}})} \\ = \ & e(g,g)^{rq_x(0)}. \end{aligned}$$

If $u_t \notin GA_x$, the user u_t cannot compute the values $e(g,g)^{rq_x(0)}$, as the exponent of D'_x in SK cannot include the inverse of the exponent K_{λ_x} of C'_x.

If $\lambda_x \notin \wedge$ or $u_t \notin GA_x$, the $DecryptNode(CT', SK, x)$ will output invalid value.

If the node x is a non-leaf node, the $DecryptNode(CT', SK, x)$ can be named from all nodes z which are children of x. For all nodes z call $DecryptNode(CT', SK, z)$ which use Lagrange coefficient to compute and obtain $e(g,g)^{rq_x(0)}$. Therefore, if the access tree T is satisfied by $\wedge$, and the user has valid memberships for each attribute group AG_i for all $\lambda_i \in \wedge$. Let $A = DecryptNode(CT', SK, R) = e(g,g)^{r_t s}$.

The user decrypts the ciphertext

$$\begin{aligned} \frac{\tilde{C}}{(e(C,D)/e(g,g)^{r_t s})} &= \frac{\tilde{C}}{(e(h^s, g^{(\alpha + r_t)/\beta})/e(g,g)^{r_t s})} \\ &= \frac{\tilde{C}}{(e(g^{\beta s}, g^{(\alpha + r_t)/\beta})/e(g,g)^{r_t s})} \\ &= \frac{Me(g,g)^{\alpha s}}{e(g,g)^{s\alpha}} \\ &= M. \end{aligned}$$

Key Update Phase

When a user comes to hold or drop an attribute, the corresponding key should be updated to avoid backward and forward secrecy on the previous or subsequent encrypted data. The key update procedure is executed by the authority when the user requests to join or leave on the attribute group. The authority receives the request, and sends the updated membership list of the attribute group to the CSS. Then the CSS receives the update request, and computes the corresponding attribute group key.

The CSS selects a random value $s' \in Z_p^*$ and a random value K'_{λ_i} which is different from the previous attribute group key $K_{\lambda_i} \neq K'_{\lambda_i}$. Then the CSS re-encrypts the ciphertext CT using the public param-

eters PP and public key PK as

$$\begin{aligned} CT' &= (T, \tilde{C}, C, C_i, C_i', \forall y \in Y\{i\} : C_y, C_y'). \\ \tilde{C} &= Me(g,g)^{\alpha(s+s')} \\ C &= h^{s+s'} \\ C_i &= g^{q_i(0)+s'} \\ C_i' &= (H_1(\lambda_i)^{q_i(0)+s'})^{K'_{\lambda_i}} \\ C_y &= g^{q_i(0)+s'} \\ C_y' &= (H_1(\lambda_y)^{q_y(0)+s'})^{K_{\lambda_y}}. \end{aligned}$$

In the other attribute groups, the attribute group keys do not necessarily need to be uploaded because they will not be affected by the membership changes.

The CSS generates a new polynomial function $f^i(x)$ with a new attribute group AG_i including a new joining user who comes to hold an attribute λ_i (or excluding a leaving user who comes to drop an attribute λ_i). The CSS generates a new header message Hdr_i with the attribute group key K'_{λ_i} as $Hdr = (g^\rho, Hdr_i, \forall y \in Y\{i\} : Hdr_y)$, where the header message Hdr_y are the same before. Finally, the CSS responds new header message and the ciphertext.

3.3 Yang et al.'s Scheme

Yang et al. [40] proposed an attribute revocation scheme in CP-ABE which utilized the access structure of linear secret sharing scheme (LSSS) to define access control in cloud storage service. Their scheme did not need to re-encrypt the ciphertext by the CSS, because they considered to be unsafe from the semi-trusted CSS re-encrypting. However, the authority needed to spend high resource of computation in their scheme. Because their key update had three parts including update key generation, secret key update and ciphertext update, the authority needed to update the ciphertext and produce new keys that include the new version key, update key, and secret key, the ciphertext and producing new keys in their scheme

Next we will describe their scheme including setup, key generation, data encryption, data decryption and key update phase.

Setup Phase

The authority chooses random values $\alpha, \beta, \Upsilon, a \in Z_p$, and generates the public parameter $PP = \{g, g^a, g^{1/\beta}, g^\beta, e(g,g)^\alpha\}$, and the master keys are $MK = \{\alpha, \beta, \Upsilon, a\}$.

For each attribute x, the authority generates a random value $v_x \in Z_p$ as the attribute version number $VK_x = v_x$. Then the authority utilizes VK_x to generate a public attribute key $PK_x = (PK_{1,x} = H_1(x)^{v_x}, PK_{2,x} = H_1(x)^{v_x \Upsilon})$.

Key Generation Phase

When a user joins the system, the authority first assigns a set of attribute S to this user according to its identity.

The authority uses the master key MK, a set of attributes S that describes the secret key, and the corresponding set of attributes the user's secret key $SK = (K = g^{\alpha/\beta} \cdot g^{(at)/\beta}, L = g^t, \forall x \in S : K_x = g^{t\beta^2} \cdot H_1(x)^{v_x t\beta})$. Finally, the authority sends SK to the user in a secure channel.

Data Encryption Phase

The data owner first divides the data $M = \{m_1, m_2, \cdots, m_n\}$ according to the logic granularity. Then it uses symmetric encryption methods to encrypt the data as the content key $k = \{k_1, k_2, \cdots, k_n\}$ where $k_i = E_K(m_i)$.

The data owner uses the public parameter PP, a set of public attribute key $\{PK_x\}$, a content key k and a LSSS access structure (TM, ρ). Let TM be a $l \times n$ matrix, where l means the number of attributes involved in the encryption. The function ρ which is associated rows of TM to attributes is a limited injective function. It first chooses a random encryption exponent $s \in Z_p$ and a random vector $\vec{v} = (s, y_2, \cdots, y_n) \in Z_p^n$, where $y_2, \cdots, y_n$ are used to share the encryption exponent s. For $i = 1$ to l, it computes $\lambda_i = \vec{v} \cdot TM_i$, where TM_i is the vector corresponding to the ith row of TM. Then it chooses random values $r_1, r_2, \cdots, r_l \in Z_p$ and computes the ciphertext $CT = (C = ke(g,g)^{\alpha s}, C' = g^{\beta s}, \forall i = 1, \cdots, l, C_i = g^{a\lambda_i}(g^\beta)^{-r_i v_\rho(i)}, D_{1,i} = H_1(\rho(i))^{v_{\rho(i)} r_i \Upsilon}, D_{2,i} = g^{r_i/\beta})$.

Data Decryption Phase

The user receives the data from the CSS. Only the attribute that the user possesses satisfies the access structure defined in the ciphertext CT, so the user can get the data component successfully. Users with different attributes will be able to decrypt different number of data components, such that they can get different granularities of information from the same data.

The user uses a ciphertext CT attached with the access structure (TM, ρ) and the secret key for a set of attribute S. The user's attribute set S satisfies the access structure and let $I \subset \{1, 2, \cdots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then it chooses a set of constants $\{w_i \in Z_p\}_{i \in I}$ and reconstructs the encryption exponent as $s = \sum_{i \in I} w_i \lambda_i$ if $\{\lambda_i\}$ are valid shares of the secret s according to TM. Then the user first

computes

$$\begin{aligned}
& \frac{e(C', K)}{\prod_{i\in I}(e(C_i, L)e(D_{2,i}, K_{\rho(i)}))^{w_i}} \\
= & \frac{e(g^{\beta s}, g^{\alpha/\beta} \cdot g^{(at)/\beta})}{\prod_{i\in I}(e(g^{a\lambda_i} H_1(\rho(i))^{-v_{\rho(i)} r_i}, g^t) \cdot e(g^{r_i/\beta}, H_1(\rho(i))^{v_{\rho(i)} t\beta}))^{w_i}} \\
= & \frac{e(g,g)^{\alpha s} e(g,g)^{sat}}{e(g,g)^{at\sum_{i\in I}\lambda_i w_i}} \\
= & e(g,g)^{\alpha s}.
\end{aligned}$$

It can then decrypt the content key $k = C/e(g,g)^{\alpha s}$. The user uses a symmetric key and the content keys to further decrypt the data $D_K(k) = m$.

Key Update Phase

1) Update Key Generation by the Authority: When these is an attribute revocation, the authority generates the update key by using the master key MK and the current version key $VK_{x'}$ of the revoked attribute x'.

   It chooses a random value $v'_{x'} \in Z_p$ where $v'_{x'} \neq v_{x'}$ and generates a new attribute version key $VK'_{x'}$.

   The authority uses $VK'_{x'}$ to computes the update key $UK_{x'} = (UK_{1,x'} = \frac{v'_{x'}}{v_{x'}}, UK_{2,x'} = \frac{v_{x'} - v'_{x'}}{v_{x'}\Upsilon}$. The authority sends the update key $UK_{x'}$ to the CSS. Then the authority also updates the public attribute key of the revoked attribute x' as

$$\begin{aligned}
PK'_{x'} &= (PK'_{1,x'}, PK_{2,x'}). \\
PK'_{1,x'} &= (PK_{1,x'})^{UK_{1,x'}} = H_1(x')^{v'_{x'}} \\
PK_{2,x'} &= (PK_{2,x'})^{UK_{1,x'}} = H_1(x')^{v'_{x'}\Upsilon}.
\end{aligned}$$

   Finally, the authority broadcasts a message to all the users that the public attribute key of the revoked attribute x' is updated.

2) Secret Key Update by Non-revoked Users: Each non-revoked user sends two components $L = g^t$ and $K_{x'}$ of the secret key SK to the authority. The authority receives these components and computes a new components $K'_{x'} = (K_{x'}/L^{\beta^2})^{UK_{1,x'}} \cdot L^{\beta^2} = g^{t\beta^2} \cdot H_1(x')^{v'_{x'} t\beta}$. Then it returns the new component $K'_{x'}$ to the non-revoked user.

   The non-revoked user's secret key is updated by replacing the component $K_{x'}$ associated with the revoked attribute x' with the new one $K'_{x'}$ as $SK' = (K, L, K_{x'}, \forall x \in S\{x'\} : K_x)$.

3) Ciphertext Update by Cloud Server: The CSS receives the update key UK_x from the authority and updates the ciphertext associated with the revoked attribute x'. The CSS uses the ciphertext CT and the update key $UK_{x'}$ to update the ciphertext $CT' = (\tilde{C} = C, \tilde{C'} = C', \forall i = 1, cdots, l : \tilde{D_{2,i}} = D_(2,i)$, if $\rho(i) \neq x' : \tilde{C_i} = C_i, \tilde{D_{1,i}} = D_{1,i}$, if $\rho(i) = x' : \tilde{C_i} = C_i \cdot (D_{1,i})^{UK_{2,x'}}, \tilde{D_{1,i}} = (D_{1,i})^{UK_{1,x'}})$.

3.4 Zu et al.'s Scheme

Zu et al. [45] proposed a new CP-ABE scheme which utilized the access structure of linear secret sharing scheme (LSSS) to define access control in cloud storage service. Their scheme had efficient revocation and fine-grained access control. Their scheme combined proxy re-encryption with CP-ABE to achieve the user and attribute revocation. In their scheme, the authority generated two secret keys of the user where one sends to the user, and the other sends to the cloud storage server. When the authority removes a user's attribute, their scheme would not affect other users' access privileges with this attribute. Finally, their scheme could reduce the load from the authority on the revocation.

Next we will describe their scheme including setup, key generation, data encryption, data re-encryption, and data decryption phase.

Setup Phase

The authority chooses random values $\alpha_1, \alpha_2, a \in Z_p$ such that $\alpha = \alpha_1 + \alpha_2 \bmod p$, and generates the public parameter $PP = \{G_1, g, H_1, e(g,g)^{\alpha}, g^a\}$, and the master keys are $MK = \{\alpha_1, \alpha_2, g^{\alpha}\}$.

Key Generation Phase

The authority uses one part of the master key α_1, a set of attributes S and chooses a random value $t \in Z_p$. The user's secret key is generated as $SK_1 = \{K = g^{\alpha_1} g^a t, L = g^t, \forall x \in S : K_x = H_1(x)^t\}$. The authority uses the other part of the master key α_2 to generate the delegation key $SK_2 = \{D_c = g^{\alpha_2}\}$ for the CSS.

Data Encryption Phase

When a data owner wants to upload its data M to the CSS for sharing, the data owner uses the public parameters PP and an LSSS access structure (TM, ρ) to encrypt a message M. Let TM be an $l \times n$ matrix, TM_i be the vector corresponding to the ith row of TM. The function ρ which is associated with rows of TM to attributes is a limited injective function.

The data owner chooses random values $r_1, r_2, \cdots, r_l \in Z_p$ and a random vector $\vec{v} = (s, y_2, \cdots, y_n) \in Z_p^n$. These elements of vector $\vec{v}$ will be used to share the encryption exponent s. For $i = 1$ to l, computes $\lambda_i = TM_i \vec{v}$. The ciphertext is published as $CT = \{\tilde{C} = Me(g,g)^{\alpha s}, C = g^s, \forall 1 \leq i \leq l, \rho(i) \in S : C_i = g^{\lambda_i} H_1(\rho(i))^{r_i}, D_i = g^{r_i}\}$ along with a description of (TM, ρ).

Data Re-encryption Phase

When a user comes to hold or drop an attribute,

the corresponding key associated with the attribute should be updated. Because the re-encryption can prevent the user from accessing the previous or subsequent re-encrypted data for backward or forward secrecy, the key associated with the attribute needs to be updated. We denote ID_i as the identity of the user i.

1) If there is no attribute revocation, the CSS uses a random $k \in Z_p$ to encrypt the delegation key g^{α_2} and the ciphertext $CT = (D'_c = (g^{\alpha_2})^k, \tilde{C} = Me(g,g)^{\alpha s}, C = g^s, C' = g^{s/k}, C'_i = g^{a\lambda_i} H_1(\rho(i))^{r_i} H_1(\rho(i))^k, D'_i = g^{r_i} g^k)$.
   The re-encrypted ciphertext $\tilde{CT} = \{CT', D'_c\}$ is then sent to the user, where $CT' = \{\tilde{C}, C, C', \{C'_i, D'_i\}_{i=1,\cdots,l}\}$.
2) If there is an attribute x' revocation from a user ID_j where ID_j means the identity of the user j, the CSS encrypts a random key $\nu_{x'} \in Z_p$ as $\tilde{C}$ under the access structure (TM, ρ) for those users $ID_i, i \neq j$ who hold the revoked attribute but not been revoked. The method of encrypting random keys and decrypting ciphertext $\tilde{C}$ is similar to that of Liang et al. scheme [35].

Then the CSS utilizes a random value $k \in Z_p$ to encrypt the delegation key g^{α_2} and the ciphertext $CT = (D'_c = (g^{\alpha_2})^k, \tilde{C} = Me(g,g)^{\alpha s}, C = g^s, C' = g^{s/k}, \forall i = 1, 2, \cdots, l, C'_i = g^{a\lambda_i} H_1(\rho(i))^{r_i} H_1(\rho(i))^k$, for $\rho(i) \neq x' : D'_i = g^{r_i} g^k$; for $\rho(i) = x : D'_i = (g^{r_i} g^k)^{1/\nu_{(\rho(i))}})$.

The re-encrypted ciphertext $\tilde{CT} = \{CT', D'_c, \tilde{C}\}$ is then sent to the users, where $CT' = \{\tilde{C}, C, C', \{C'_i, D'_i\}_{i=1,\cdots,l}\}$.

Data Decryption Phase

A user receives the ciphertext $\tilde{CT}$ for access structure (TM, ρ), and uses the private key SK_1 for a set of attributes S to decrypt:

1) If there is no attribute revocation, the user computes

$$A = \frac{\prod_{i \in I} e(C'_i, L)^{w_i}}{\prod_{i \in I} e(D'_i, K_{\rho(i)})^{w_i}} = e(g,g)^{ats}.$$

   The user decrypts the ciphertext

$$\frac{\tilde{C} \cdot A}{e(C', D'_c) e(C, K)} = \frac{Me(g,g)^{\alpha s} \cdot e(g,g)^{ats}}{e(g^s, g^{\alpha_2 k} e(g^s, g^{at}))} = M.$$

2) If there is an attribute x' revocation from a user ID_j. The user $ID_i, i \neq j$ holds the revoked attributes S and satisfies with the access structure (TM, ρ), then the user decrypts $\tilde{C}$ using SK_1 and obtains $\nu_{x'}$ to update the secret key $K_{x'}$ as $K_{x'} = (H(x')^t)^{\nu_{x'}}$. Otherwise, he/she cannot get the updated secret key $K_{x'}$.

The first step of decryption of $\tilde{CT}$ proceeds in the following: for $\rho(i) \neq x' : B_i = \frac{e(C'_i, L)^{w_i}}{e(D'_i, K_{\rho(i)})^{w_i}} = e(g,g)^{at\lambda_i w_i}$, for $\rho(i) = x' : B_i = \frac{e(C'_i, L)^{w_i}}{e(D'_i, \tilde{K_{\rho(i)}})^{w_i}} = e(g,g)^{at\lambda_i w_i}$, $A = \prod_{i \in I} B_i = e(g,g)^{ats}$. The user decrypts the ciphertext $\frac{\tilde{C} \cdot A}{e(C', D'_c) e(C, K)} = \frac{Me(g,g)^{\alpha s} \cdot e(g,g)^{ats}}{e(g^s, g^{\alpha_2 k}) e(g^s, g^{at})} = M$.

4 Analysis

In the section, we will analyze these schemes [13, 14, 40, 45] which contain functional requirement, security and performance. And we also use the tables to present a corresponding requirement in each scheme.

4.1 Functional Evaluation

In Table 2, we will analyze several functional requirements: fine-grained access control, scalability, user accountability, user revocation, collusion resistant, forward secrecy and backward secrecy in the representative approaches. Almost schemes can achieve these functional requirement including data confidentiality, fine-grained access control, user revocation, collusion resistant, forward secrecy and backward secrecy. In K. Yang et al.'s scheme, when an attribute is revoked, non-revoked users need to update their secret keys. Therefore, their scheme did not satisfy the scalability.

4.2 Performance Evaluation

We will analyze four phases: setup phase, key generation phase, data encryption phase, data re-encryption phase; data decryption phase and user revocation phase in the four entities include data owner, user (the group user), cloud storage server (CSS) and the authority. Before we analyze the performance evaluation, first we introduce the notations in Table 3.

In Table 4, we analyze four schemes how to execute a setup phase. Hur's scheme needs the CSS to generate the public key, the master key and another key because they considered the key escrow problem. K. Yang et al.'s scheme spent more computing resources.

In Table 5, we analyze four schemes how to execute a key generation phase. Because the CSS executes partly computation, the authority could reduce computation in Hur's scheme. However, Zu et al.'s scheme needed lower computation in these schemes when a set of attributes are smaller.

In Table 6, we analyze four schemes how to execute a data encryption phase. K. Yang et al.'s scheme needed more computing resources, but they did not execute re-encryption phase. Hur and Noh's scheme and Hur's scheme needed less computing resource.

Table 2: Comparison of functional requirements

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
Data confidentiality	Yes	Yes	Yes	Yes
Fine-grained access control	Yes	Yes	Yes	Yes
Scalability	Yes	Yes	No	Yes
User revocation	Yes	Yes	Yes	Yes
Collusion resistant	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes
Backward secrecy	Yes	Yes	Yes	Yes

Table 3: Notations

Notation	Significance
T_{sym}	The computing time of symmetric encryptions
T_{Ge}	The computing time of exponentiation in group operation
T_B	The computing time of bilinear pairing
T_{Mul}	The computing time of multiplication
T_{Div}	The computing time of division
T_{Add}	The computing time of addition
T_{Sub}	The computing time of subtraction
T_{GM}	The computing time of multiplication in group operation
T_h	The computing time of hash function
$\wedge$	A set of attributes
i	A set of revoked attributes
m	The number of users in the group

Table 4: Comparison of computation in the setup phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
CSS		$3T_{Ge}$		
Authority	$2T_{Ge}$	$1T_{Ge}$	$5T_{Ge} + 2T_h$	$3T_{Ge} + 1T_A$

Table 5: Comparison of computation in the key generation phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
CSS		$3T_{Ge} + T_h$		
Authority	$2T_{Ge} + T_{GM}$ $+ \wedge (3T_{Ge} + T_{GM} + T_h)$	$T_{Ge} + T_{Mul}$ $+ \wedge (3T_{Ge} + T_{GM} + T_h)$	$4T_{Ge} + T_{GM}$ $+ \wedge (5T_{Ge} + T_{GM} + T_h)$	$4T_{Ge} + T_{GM}$ $+ \wedge (T_{Ge} + T_h)$

Table 6: Comparison of computation in the data encryption phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
Data owner	$2T_{Ge} + T_{GM}$ $+ \wedge (2T_{Ge} + T_h)$	$2T_{Ge} + T_{GM}$ $+ \wedge (2T_{Ge} + T_h)$	$T_{sym} + 2T_{Ge} + T_{GM}$ $+ \wedge (7T_{Ge} + T_{GM} + T_h)$	$2T_{Ge} + T_{GM}$ $+ \wedge (3T_{Ge} + T_{GM} + T_h)$

In Table 7, we analyze four schemes how to execute a data re-encryption phase. Hur and D. Noh's scheme needed less computing resource in these scheme. K. Yang et al. did not execute data re-encryption.

In Table 8, we analyze four schemes how to execute a data decryption phase. Hur's scheme needed to spend more computing resource because they considered details on the decryption. Zu et al.'s scheme was better in these scheme because they need less computing resources.

In Table 9, we analyze four schemes how to execute a key update phase. K. Yang et al.'s scheme needed to compute the CSS and the authority together. Although Zu et al.'s scheme did not support key update, their schemes executed re-encryption in the situation of attribute revocation. Hence, we describes the situation of attribute revocation in key update.

5 Conclusion and Future Work

Conclusion

Although cloud data storage has many advantages, it also bring many challenges. When the cloud service provider provides a semi-trusted cloud server, it may steal clients' data which is serious issues. Therefore, data confidentiality and access control are important issues in cloud storage. Then cloud data can share own data with other in cloud platform. Therefore, the access controls which users to share the data together, and a user leaves the access privilege of the data. Although there are many kinds of access control schemes, they have to apply the restriction of cloud environment. Because users store data in the cloud storage, they cannot control their data. Attribute-based encryption is a promising scheme in data security which can limit the data to access control. Ciphertext-policy attribute-based encryption is an applicable scheme in the cloud data storage which encrypts the data and defines access structure from the user.

Therefore, we survey the previous researches of attribute-based access control with user revocation in the cloud. Then we collect and explain basic requirements in the mechanism. We analyze these approaches by using function and performance evaluation. Finally, in this paper, we provide the future development of CP-ABE with user revocation.

Future Work

For future developments, we will focus on the following areas of particular interest. Efficiency: The authority needs to generate every user' key and other computing. When a lot of users constantly change in access control, it will cause the authority to spend more computing resources. Therefore, how to avoid frequently change in the key update is an important issue. Security: the user's key will be a challenge because the key is generated by the authority. Because the key distribution is constructed in a secure channel, how to design a public channel scheme is an important issue.

References

[1] M. Armbrust, et al., "A view of cloud computing", *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] N. Attrapadung, H. Imai, "Conjunctive broadcast and attribute-based encryption", in *Proceedings of the 3rd International Conference on Pairing-Based Cryptography*, pp. 248–265, 2009.

[3] N. Attrapadung, H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes", in *Cryptography and Coding*, pp. 278–300, 2009.

[4] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, California, USA, May 20-23, 2007.

[5] A. Boldyreva, V. Goyal, V. Kumar, "Identity-based encryption with efficient revocation", in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08)*, pp. 417–426, 2008.

[6] V. Bovović, D. Socek, R. Steinwandt, V. I. Villányi, "Multi-authority attribute-based encryption with honest-but-curious central authority", *International Journal of Computer Mathematics*, vol. 89, no. 3, pp. 268–283, 2012.

[7] M. Chase, "Multi-authority attribute based encryption", in *Proceedings of the 4th Conference on Theory of Cryptography*, pp. 515–534, 2007.

[8] M. Chase, S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption.", in *Proceedings of the 16th ACM conference on Computer and Communications Security (CSS'09)*, pp. 121–130, 2009.

[9] V. Goyal, O. Pandey, A. Sahai, B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, Alexandria, Virginia, USA, 2006.

[10] S. Guo, Y. Zeng, J. Wei, Q. Xu, "Attribute-based re-encryption scheme in the standard model", *Wuhan University Journal of Natural Sciences*, vol. 13, no. 5, pp. 621–625, 2008.

[11] J. Han, W. Susilo, Y. Mu, J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption", *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[12] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, no. 6, pp. 98–115, 2015.

Table 7: Comparison of computation in the data re-encryption phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
CSS	$\wedge(T_{sym}+T_{Ge})$	$2T_{Ge}+T_h+T_B+\wedge(T_{Ge}+T_{GM})$	No	$2T_{Ge}+\wedge(2T_{Ge}+1T_h)$

Table 8: Comparison of computation in the data decryption phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
User	$\wedge(T_B+T_{Ge})$ $+T_B+T_{GM}$	$T_h+2T_B+T_{GM}(m+2)$ $+\wedge(T_{Ge}+T_B)+2T_{Ge}$	$\wedge(T_B+T_M)$ $+T_{GM}+T_{sym}$	$\wedge T_B+3T_{GM}+2T_B$ $+T_h+2T_{Ge}$

Table 9: Comparison of computation in the key update phase

	Hur and Noh [14]	Hur [13]	Yang et al. [40]	Zu et al. [45]
CSS	$5T_{Ge}+T_h+T_{GM}$ $+(\wedge-i)(2T_{Ge}+2T_{GM}$ $+T_h)+\wedge T_{sym}$	$6T_{Ge}+T_h+T_{GM}$ $+(\wedge-i)(2T_{Ge}+2T_{GM}$ $+T_h)+\wedge(T_{GM})$	$(\wedge-i)(T_{GM}+2T_{Ge})$	$2T_{Ge}+(\wedge-i)(6T_{Ge}$ $+T_{GM}+2T_h)$
Authority			$2T_{Div}+T_sub+2T_h$ $+4T_{Ge}+T_{Mul}+T_{GM}$	

[13] J. Hur, "Improving security and efficiency in attribute-based data sharing", *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.

[14] J. Hur, D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[15] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application", in *Information Security Applications*, pp. 309–323, 2009.

[16] P. Junod, A. Karlov, "An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies", in *Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management*, pp. 13–24, 2010.

[17] R. Kui, W. Cong, W. Qian, "Security challenges for the public cloud", *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[18] A. Lewko, B. Waters, "Decentralizing attribute-based encryption", in *Advances in Cryptology (EUROCRYPT'11)*, pp. 568–588, Springer Berlin Heidelberg, 2011.

[19] J. Li, X. Chen, C. Jia, W. Lou, "Identity-based encryption with outsourced revocation in cloud computing", *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 425–437, 2015.

[20] J. Li, Q. Huang, X. Chen, S. S. Chow, D. S. Wong, D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability", in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 386–390, 2011.

[21] J. Li, K. Ren, K. Kim, "A2BE: Accountable attribute-based encryption for abuse free access control", *IACR Cryptology ePrint Archive*, vol. 20092009, pp. 118, 2009.

[22] J. Li, K. Ren, B. Zhu, Z. Wan, "Privacy-aware attribute-based encryption with user accountability", in *Information Security*, pp. 347–362, 2009.

[23] K. Liang, L. Fang, W. Susilo, D. Wong, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security", in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS'13)*, pp. 552–559, 2013.

[24] X. Liang, Z. Cao, H. Lin, J. Shao, "Attribute based proxy re-encryption with delegating capabilities", in *Proceedings of the 4th ACM International Symposium on Information, Computer, and Communications Security (ASIACCS'09)*, pp. 276–286, 2009.

[25] X. Liang, R. Lu, X. Lin, X. S. Shen, *Ciphertext Policy Attribute Based Encryption with Efficient Revocation*, Technical Report, University of Waterloo, 2010.

[26] H. Lin, Z. Cao, X. Liang, J. Shao, "Secure threshold multi authority attribute based encryption without a central authority", *Information Science*, vol. 180, no. 13, pp. 2618–2632, 2010.

[27] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, T. H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles", in *Proceedings of the 16th European Con-*

ference on Research in Computer Security (ESORICS'11), pp. 278–297, Springer Berlin Heidelberg, 2011.

[28] S. Luo, J. Hu, Z. Chen, "Ciphertext policy attribute-based proxy re-encryption", in *Information and Communications Security*, pp. 401–415, 2010.

[29] D. Naor, M. Naor, J. Lotspiech, "Revocation and tracing schemes for stateless receivers", in *Advances in Cryptology (CRYPTO'01)*, pp. 41–62, Springer, 2001.

[30] R. OstrovOstrovsky, A. Sahai, B. Waters, "Attribute-based encryption with non-monotonic access structures", in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pp. 195–203, 2007.

[31] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, "Secure attribute-based systems", in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 99–112, 2006.

[32] S. Rafaeli, D. Hutchison, "A survey of key management for secure group communication", *ACM Computing Surveys*, vol. 35, no. 3, pp. 309–329, 2003.

[33] A. Sahai, B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, LNCS 3494, pp. 457–473, Springer, 2005.

[34] H. J. Seo, H. Kim, "Attribute-based proxy reencryption with a constant number of pairing operations", *Journal of Information and Communication Convergence Engineering*, vol. 10, no. 1, pp. 53–60, 2012.

[35] J. Staddon, P. Golle, M. Gagne, P. Rasmussen, "A content-driven access control system", in *Proceedings of the 7th ACM Symposium on Identity and Trust on the Internet*, pp. 26–35, 2008.

[36] S. Subashini, V. Kavitha, "A survey on security issues in service delivery models of cloud computing", *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[37] Y. Wang, K. Chen, Y. Long, Z. Liu, "Accountable authority key policy attribute-based encryption", *Science China Information Sciences*, vol. 55, no. 7, pp. 1631–1638, 2012.

[38] X. Xie, H. Ma, J. Li, X. Chen, "New ciphertext-policy attribute-based access control with efficient revocation", in *Information and Communication Technology*, LNCS 7804, pp. 373–382, Springer, 2013.

[39] Z. Xu, R. Holloway, K. M. Martin, "Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage", in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12)*, pp. 844–849, 2012.

[40] K. Yang, X. Jia, *Security for Cloud Storage Systems*, Springer, New York, 2014.

[41] K. Yang, X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2014.

[42] S. Yu, K. Ren, W. Lou, and J. Li, "Defending against key abuse attacks in KP-ABE enabled broadcast systems", in *5th International ICST Conference on Security and Privacy in Communication Networks (SecureComm'09)*, pp. 311–329, 2009.

[43] S. Yu, C. Wang, K. Ren, W. Lou, "Attribute based data sharing with attribute revocation", in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, pp. 261–270, 2010.

[44] D. Zissis, D. Lekkas, "Addressing cloud computing security issues", *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[45] L. Zu, Z. Liu, J. Li, "New ciphertext-policy attribute-based encryption with efficient revocation", in *Proceedings of the 2014 IEEE International Conference on Computer and Information Technology (CIT'14)*, pp. 281–287, 2014.

Chih-Wei Liu received his M.S. in Soil And Water Conservation from National Chung Hsiung University, Taichung, Taiwan, ROC, in 2008. He is currently pursuing the Ph.D. degree from Computer Science and Information Engineering Department of Asia University, Taichung, Taiwan. His research interests include information security, cloud computing, and information law.

Wei-Fu Hsien received his B. S. in Department of Information Management from National Kaohsiung Marine University, Kaohsiung, Taiwan, ROC, in 2013. He is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. His research interests include security and privacy of cloud computing, and applied cryptography.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. From 1994 to 2004, Dr. Yang was an associate professor in the Department of Computer Science and Information Engineering, Chaoyang University of Tech- nology. Currently, he is a professor in the Department of Management Information Systems, National Chung Hshing University. His research interests include network security, mobile computing, and distributed system.

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.

A Joint Random Secret Sharing Scheme with Public Verifiability

Zhenhua Chen[1], Shundong Li[2], Qiong Huang[3], Jianhua Yan[4], Yong Ding[5]
(Corresponding author: Zhenhua Chen)
School of Computer Science and Technology, Xi'an University of Science and Technology[1]
Shaanxi, China
(Email: chenzhenhua@snnu.edu.cn)
School of Computer Science, Shaanxi Normal University[2]
College of Mathematics and Informatics, South China Agricultural University[3]
State Key Laboratory of Networking and Switching Technology, Beiyou University[4]
Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology[5]
(Received July 12, 2015; revised and accepted Oct. 15 & Nov. 2, 2015)

Abstract

In this paper, we propose a joint random secret sharing scheme with public verifiability. It is practical in distributed environment. Utilizing additive homomorphism, a random secret will be corporately constructed by some participants, which avoids the need for a mutually trusted dealer. In addition, we explore the technique of homomorphic verification and that of bilinear pairing to allow each participant to publicly verify whether the received shares are consistent. The verification process in our scheme is unconditionally secure and non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof, which is simple and higher efficient compared with previously known. Lastly, as an applied example of our work, we present how our techniques can be applied to handle dynamic node-join in mobile ad hoc network.

Keywords: Additive homomorphism, bilinear pairing, joint random secret sharing, public verifiability, unconditionally secure

1 Introduction

1.1 Background and Motivation

Secret sharing, invented independently by Shamir [24] and Blakley [2] in 1979, is a fundamental cryptographic primitive that has been found useful in numerous applications such as witness encryption [13], access control [15], secure group communication [10], secure information communication [16], and cloud storage [4]. In a (k, n) secret sharing scheme (SSS), a secret s is distributed to n participants by a dealer in such a way that any k participants or more can reconstruct the secret s but participants less than k learn nothing about s with their shares. However, there are several common drawbacks in the existing SSS [2, 24].

1) A dishonest dealer may distribute a fake share to a certain participant, and then the participant would never obtain the true secret subsequently.

2) A malicious participant may submit a fake share, which makes it the only one who gets to reconstruct the true secret after observing the shares of honest participants, whereas the honest cannot obtain the true secret.

3) A mutually trusted dealer must be needed for generating and distributing each share to the corresponding participant secretly.

4) It is required that there exists a private channel between the dealer and each participant during the share distribution phase.

To address the cheating problems in 1) and 2), a verifiable algorithm can be added to SSS, which is called verifiable secret sharing scheme (VSSS) [3] and can be used to verify whether the shares are consistent. VSSS further are investigated by many other researchers. Feldman [5] and Pederson [22] proposed a VSSS based on Shamir's scheme, respectively, which can effectively detect cheating of the participants or the dealer. The security of verifiability in VSSS can be classified into two types: computational security and unconditional security. The verification process in the former is unconditionally secure, whereas that in the latter is computationally secure. More specifically, it is based on the hardness of solving the discrete logarithm.

In many general schemes including those discussed above, there is a dealer whose function is to distribute the shares among participants. Nevertheless, it is difficult to

find a trustworthy person or organization as the dealer in the real world. In addition, the mutually trusted dealer has too much rights and is easy to suffer from the adversary's attacks. To deal with the drawbacks in 3), Ingemarsson and Simmons [11] introduced a new type of SSS without the assistance of a mutually trusted dealer, which is called joint random secret sharing scheme (JRSSS). The basic idea of JRSSS is that each participant also acts as a dealer to select a sub-secret and to generate sub-shares for others. The master secret is the summation of all sub-secrets. However, there is one potential problem in their design, that is, each participant needs to keep n sub-shares secretly and the number of shares kept by each participant is proportional to the number of participants. Therefore, the cost of storage and management of shares are expensive. Pederson [23] proposed a solution to overcome this problem. According to the property of additive homomorphism defined in [1], each participant only needs to keep one master share secretly.

Based on Pederson's approach [23], Harn and Lin [7] introduced a new notion of strong k-consistency of shares and proposed a verifiable JRSSS (VJRSSS), which enables participants to verify their shares whether to satisfy the security requirements of a (k, n) SSS. However, in their scheme, shareholders need to utilize 100 verification polynomials to verify the strong t-consistency of master shares, which makes the verification more complicated and spends too much time. After that, numerous VJRSSS were proposed to reduce the computational complexity. In 2012, Liu et al. [19] updated the scheme of Harn and Lin [7], in which shareholders utilize the sub-polynomials of master secret to construct a verification polynomial and use it to verify master shares. In 2013, Mahmoud [20] constructed a polynomial differential-based VJRSSS, in which they calculate the t-th derivative of polynomials and apply Shamir's SSS to generate and distribute the sub verification shares and use Pedersen's SSS to find the master verification shares. However, this scheme seems not to guarantee the strong t-consistency, i.e., it cannot detect the fact of cheating.

Despite the research results, we observe that participants in the existing VJRSSS can only verify their own shares rather than others and only detect the fact of cheating but not identify who are the cheaters. Though Kaya et al. [12] designed a VJRSSS which can identify who are cheaters, their scheme does not hold public vitrifiability, i.e., the shares cannot be verified by anyone. Stadler [25] introduced a publicly verifiable secret sharing (PVSS) scheme, in which not only the participants can verify the validity of their shares, but also any one can do it from the public information without revealing shares. Note that unconditional secrecy is not possible in a PVSS scheme since the encrypted shares are sent by public channels, namely, no private channels between the dealer and each participant are assumed. So, the PVSS schemes overcome the drawbacks in 4). Recently, Villar and Heidarvand [8] construct a PVSS scheme using pairing, the verification of which is unconditionally secure and efficient since the verification process does not depend on any computational assumption and is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof. Nevertheless, their scheme requires a mutually trusted dealer to generate and distribute shares, which is impractical in a distributed scenario.

1.2 Our Contribution

In this paper, to solve the aforementioned problems, we first provide a joint random secret sharing scheme (JRSSS) with public verifiability. Our scheme is based on the technique of Pedersen's (k, n) SSS [23] and that of Villar et al.'s scheme [8]. However, the techniques in [8, 23] cannot be used in building our JRSSS with public verifiability directly. It is because that the share in verification equality in [8] is just only one, while in our scheme, that in verification equality is the summation of n sub-shares. In order to take advantage of these techniques in [8], we make a modification on their scheme with homomorphic verification.

We employ additive homomorphism to avoid the use of a mutually trusted dealer who selects and distributes the private shares, and explore homomorphic verification and the bilinear pairing to allow anyone to publicly verify whether the shares are consistent. To the best of our knowledge, our scheme first provides a distributed secret sharing scenario with public verifiability.

The primary advantages of our scheme are summarized as follows.

- Cheater identification: our scheme can not only detect the fact of cheating but also identity who are the cheaters.
- Unconditionally secure verifiability: the verifiability of our scheme does not depend on any computational assumption.
- Non-interactive verification: our scheme is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof.
- *Homomorphic property*: our scheme enjoys homomorphic property compatible with public verifiability.

1.3 Roadmap

In Section 2, we briefly review the related preliminary. In Section 3, we present our scheme in detail. The homomorphic property of our scheme is discussed in Section 4. Scheme analysis is provided in Section 5, while in Section 6, a performance comparison of our scheme with previously known is illustrated. An application of our scheme is showed in Section 7. Finally, in Section 8, we summarize our works.

2 Preliminary

2.1 Review of Pedersen's JRSSS

In this section, we review a joint random secret sharing scheme (JRSSS) originally proposed by Pedersen [23], in which each participant also acts as a dealer. Each participant P_i selects a random sub-secret s_i independently and the master secret s can be constructed cooperatively with the help of the homomorphism property [1], where $s = \sum_{i=1}^{n} s_i$. We now describe this scheme as follows.

Share Generation.

1) Sub-secret generation:
Each participant P_i selects a random secret s_i called as sub-secret.
2) Sub-share generation:
For each sub-secret s_i, the participant P_i selects a random polynomial $f_i(x)$ of degree $k-1$ such that $s_i = f_i(0)$ and uses Shamir's (k, n) SSS to generate sub-shares s_{ij} such that $s_{ij} = f_i(j)$. Later, P_i sends each s_{ij} to other participant P_j secretly, for $j = 1, 2, \cdots, n$.

Secret Reconstruction.

1) Master share generation:
Each participant P_j with n sub-shares s_{ij} for $i = 1, 2, \cdots, n$, computes the master share s'_j as $s'_j = \sum_{i=1}^{n} s_{ij} = \sum_{i=1}^{n} f_i(j)$.
2) Master secret reconstruction:
With knowledge of any k master shares $s'_1, \cdots, s'_k$, the master secret s can be reconstructed using the Lagrange interpolating formula:

$$\begin{aligned} s &= \sum_{j=1}^{k} s'_j \lambda_j \\ &= \sum_{j=1}^{k} (\sum_{i=1}^{n} s_{ij} \lambda_j) \\ &= \sum_{i=1}^{n} s_i, \end{aligned}$$

where $\lambda_j = \prod_{j \neq i} \frac{i}{i-j}$.

2.2 The Homomorphic Property

The homomorphic property of the secret sharing scheme was introduced by Benaloh [1]. We say that a SSS has the homomorphic property if the sum of the shares of two secrets s_1 and s_2 sent to the participants are shares of the sum of secrets $s_1 + s_2$. Therefore, the participants are able to recover the sum of secrets only knowing the shares from s_1 and s_2.

Let S be the domain of a secret and T be the domain of the shares corresponding to the secret. $F_A : T^k \to S$ is an induced function of the (k, n) SSS for each $A \subset \{1, 2, \cdots, n\}$ with $|A| = k$. This function defines the sub-secret s_i based on k sub-shares $s_{i1}, s_{i2}, \cdots, s_{ik}$, namely, $s_i = F_A(s_{i1}, s_{i2}, \cdots, s_{ik})$. Following theorem proves that each participant only needs to keep one master share secretly and then the master secret can be reconstructed based on any k master shares or more according to the property of additive homomorphism.

Theorem 1. *With knowledge of any k master shares or more, participants can reconstruct the master secret using Shamir's secret reconstruction algorithm according to the property of additive homomorphism.*

Proof.

$$\begin{aligned} s_1 &= F_A(s_{11}, \cdots, s_{1k}) \\ s_2 &= F_A(s_{n1}, \cdots, s_{nk}) \\ &\vdots \\ s_n &= F_A(s_{n1}, \cdots, s_{nk}). \end{aligned}$$

Then, we have

$$\begin{aligned} s &= s_1 + \cdots + s_n \\ &= F_A(s_{11}, \cdots, s_{1k}) + \cdots + F_A(s_{n1}, \cdots, s_{nk}) \\ &= F_A((s_{11} + \cdots + s_{n1}), \cdots, (s_{1k} + \cdots + s_{nk})) \\ &= F_A(\sum_{i=1}^{n} s_{i1}, \cdots, \sum_{i=1}^{n} s_{ik}) \\ &= F_A(s'_1, \cdots, s'_k). \end{aligned}$$

Using Lagrange interpolation, the master secret s can be reconstructed with k master shares $s'_1, \cdots, s'_k$ where $s'_i = \sum_{j=1}^{n} s_{ji}$ for $i = 1, \cdots, k$.

In the case of a SSS with public verifiability, we say that such a scheme has homomorphic property when, beside all above, the verification of the shares of the new secret $s_1 + s_2$ can be done from the broadcasted public information about s_1 and s_2. □

2.3 Bilinear Pairing

Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with the same prime order q where g is a generator of group $\mathbb{G}_1$. A bilinear pairing e is a function defined by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. For all $a, b \in \mathbb{Z}_q^*$, $P, Q \in \mathbb{G}_1$, we say e is an admissible bilinear map if the function e satisfies the following three conditions:

1) Bilinear: $e(g^a, g^b) = e(g, g)^{ab}$.
2) Non-degenerate: $e(g, g) \neq \mathbf{1}$.
3) Computable: $e(P, Q)$ is efficiently computable.

2.4 Related Complexity Assumptions

For security analysis of our proposed scheme, we summarize some important security problems and assumptions as follows.

- **Computational Diffie-Hellman (CDH) problem**: Given $g, g^a, g^b \in \mathbb{G}_1$ for some $a, b \in \mathbb{Z}_q^*$, the CDH problem is to compute $g^{ab} \in \mathbb{G}_1$.
- **CDH assumption**: No probabilistic polynomial time (PPT) algorithm can solve the CDH problem with a non-negligible probability.
- **Bilinear Diffie-Hellman (BDH) problem**: Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some $a, b, c \in \mathbb{Z}_q^*$, the BDH problem is to compute $e(g,g)^{abc} \in \mathbb{G}_2$.
- **BDH assumption**: No PPT algorithm can solve the BDH problem with a non-negligible probability.

3 Proposed Scheme

In this section, we present a joint random secret sharing scheme with public verifiability. Let $s_i \in \mathbb{Z}_q^*$ be a random sub-secret selected by each participant P_i for $i = 1, 2, \cdots, n$. A random secret s is recovered cooperatively by any k participants or more where $s = e(h^{\sum_{p_i \in A} s_i}, h)$ and A is the set of participants whose shares all are verified correctly.

Our scheme consists of three algorithms: share generation, share verification, and secret reconstruction. The concrete construction is illustrated as follows.

Share Generation.

1) Setup: Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with the same prime order q where q g, h are two independently generators of group $\mathbb{G}_1$. A bilinear pairing e is a function defined by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The public parameters $Param = (\mathbb{G}_1, \mathbb{G}_2, q, g, h, e)$ was agreed and published by all participants. Every participant publishes his public key $y_i = h^{x_i}$ and withholds the corresponding secret key $x_i \in \mathbb{Z}_q^*$.
2) Sub-secret generation: Each participant P_i selects a random sub-secret $s_i \in \mathbb{Z}_q^*$ independently.
3) Sub-share generation: P_i chooses a random polynomial $f_i(x) = \sum_{l=0}^{k-1} a_{il} x^l \bmod q$ where $a_{i0} = f_i(0) = s_i$, and uses Shamir's (k, n) SSS to generate sub-shares s_{ij} for other participant P_j such that $s_{ij} = f_i(j)$ for $j = 1, 2, \cdots, n$. After that, P_i broadcasts the commitments $C_{il} = g^{a_{il}}$ for $0 \le l \le k-1$. Later, he computes and publishes the encryption $Y_{ij} = (y_j)^{s_{ij}}$ of each sub-share s_{ij} to other participant P_j for $j = 1, 2, \cdots, n$.

   Finally, each participant P_j receives n encryptions Y_{ij} for $i = 1, 2, \cdots, n$.

Share Verification.

Any verifier can check whether each encryption Y_{ij} received by participant P_j are consistent with sub-share s_{ij} by means of checking the equation $e(\prod_{l=0}^{k-1} C_{il}^{j^l}, y_j) = e(g, Y_{ij})$ for $i = 1, 2, \cdots, n$.

Secret Reconstruction.

1) Master share generation: Let A be the set of participants whose shares all are verified correctly. Using his own secret key x_j, every participant P_j in the set A decrypts Y_{ij} as $Y_{ij}^{x_j{}^{-1}} = h^{s_{ij}} = h^{f_i(j)}$. Then, P_j computes $s'_j = \prod_{p_i \in A} h^{f_i(j)} = h^{\sum_{p_i \in A} f_i(j)}$ and saves s'_j as his master share.
2) Master share verification: The master share of participant $P_j \in A$ can be verified by others with the following verification equation:
$$e(s'_j, y_j) = \prod_{p_i \in A} e(Y_{ij}, h).$$
3) Master secret reconstruction: After the verification, then for an arbitrary subset $B \subseteq A$ consisting of k participants whose correct master shares have pooled, every participant in B can get master secret s by the following Lagrange interpolation:
$$\begin{aligned} s &= \prod_{j=1, p_j \in B}^{k} = e(s'_j, h)^{\lambda_j} \\ &= \prod_{j=1, p_j \in B}^{k} e(h,h)^{\sum_{p_i \in A} f_i(j)\lambda_j} \\ &= e(h,h)^{\sum_{p_i \in A}(\sum_{j=1, p_j \in B}^{k} f_i(j)\lambda_j)} \\ &= e(h,h)^{\sum_{p_i \in A} f_i(0)} \\ &= e(h,h)^{\sum_{p_i \in A} s_i}, \end{aligned}$$
   where $\lambda_j = \prod_{p_j \in B, j \neq i} \frac{i}{i-j}$.

4 The Homomorphic Property

Let $f_1(j)$ and $f_2(j)$ be the sub-shares of sub-secrets s_1 and s_2 for participant P_j, respectively. Following the idea from [1], we say that our scheme has the homomorphic property since Shamir's scheme also has it. So we have $f_1(j) + f_2(j)$ be the master share of the sum sub-secret $s_1 + s_2$. In relation to the verification process, if the elements (g, h, y_j, Y_{1j}) and (g, h, y_j, Y_{2j}) are used in the verifications of the sub-shares of s_1 and s_2, respectively, namely, $e(h^{f_1(j)}, y_j) = e(Y_{1j}, h)$ and $e(h^{f_2(j)}, y_j) = e(Y_{2j}, h)$, then it is easy to prove the homomorphic verification of master share of the sum sub-secret $s_1 + s_2$ if the equality
$$e(h^{f_1(j)+f_2(j)}, y_j) = e(Y_{1j}, h) e(Y_{2j}, h)$$
is satisfied.

Note that the property of homomorphic verification is not achieved if the protocol makes use of a typical zero knowledge proof in the verification process.

5 Scheme Analysis

5.1 Correctness

The correctness of this scheme means that:

1) A honest participant can always pass the verification procedure in both share generation phase and secret reconstruction phase;

2) At least t honest participants are always able to recover an unique master secret.

It is straightforward to check these requirements for the above protocol.

5.2 Verification of Scheme

In this section, we show that the participants in the protocol must behave honestly or will be detected. More precisely, on the one hand, the participants must be honest in the share generation phase and, on the other hand, the participants must be honest in the secret reconstruction phase.

5.2.1 Verification of the Share Generation

In the share generation phase, if P_i passes the verification procedure, then any qualified sets of k honest participants will reconstruct the same sub-secret s_i.

Theorem 2. *If P_i passes the verification process in share generation phase, then there exists a unique polynomial $f_i(x)$ such that the encrypted share of participant P_j is $Y_j = y_j^{f_i(j)}$ for $1 \le j \le n$, i.e., P_i must be honest.*

Proof. Assume that the encrypted sub-share of the participant P_j sent by P_i is equal to $Y_{ij} = y_j^{\bar{s_{ij}}}$. If P_i passes the equation $e(\prod_{l=0}^{k-1} C_{il}^{j^l}, y_j) = e(g, Y_{ij})$, then by the definition $C_{il} = g^{a_{il}}$, we follow $e(g, y_j)^{f_i(j)} = e(g, y_j)^{\bar{s_{ij}}}$, which leads to $\bar{s_{ij}} = f_i(j)$ for $1 \le j \le n$. Hence, the uniqueness of $f_i(x)$ will be reconstructed by any qualified sets of k correct sub-shares and then the same sub-secret $s_i = f_i(0)$ will be recovered. □

5.2.2 Verification of the Secret Reconstruction

In the share reconstruction phase, if some participant gives a different master share, then it means one of the sub-shares will be decrypted incorrectly. Otherwise, the master share should derive from n sub-shares decrypted correctly.

Theorem 3. *If P_j passes the verification process in the secret reconstruction phase, then for any i, $h^{f_i(j)} = Y_{ij}^{x_j{}^{-1}}$, where x_j is the secret key of P_j, i.e., P_j must be honest.*

Proof. Suppose $P_j \in A$ sends a different master share $\bar{s_j'} = h^{\sum_{p_i \in A} f_i(j)}$ where one of sub-shares $h^{f_i(j)}$ is decrypted by another secret key $\bar{x_j}$, namely, $h^{f_i(j)} = Y_{ij}^{\bar{x_j}}$. If any other participant accepts P_j's master share, then the following verification equality holds in the share reconstruction phase:

$$\begin{aligned}
e(s_j', y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
e(h^{\sum_{p_i \in A} f_i(j)}, y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(h^{f_i(j)}, y_j) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(Y_{ij}^{\bar{x_{ij}}}, h^{x_{ij}}) &= \prod_{p_i \in A} e(Y_{ij}, h) \\
\prod_{p_i \in A} e(Y_{ij}, h)^{\bar{x_{ij}} x_{ij}} &= \prod_{p_i \in A} e(Y_{ij}, h).
\end{aligned}$$

Thus, the above equality results in $\bar{x_j} x_j = 1$ and then $\bar{x_j} = x_j^{-1}$, which means that the sub-share $h^{f_i(j)}$ is decrypted correctly by his secret key x_j^{-1}. It follows that if any other participant accepts the master share of P_j, then the master share should derive from n sub-shares decrypted correctly. □

Note that in our scheme, the validity of shares can be publicly verified without leaking the privacy of shares and secret in the share verification phase. Furthermore, the verification process does not depend on any computational assumption and is non-interactive without using Fiat-Shamir technique or any additional zero knowledge proof.

The results in this section are summarized in the following theorem.

Theorem 4. *The verification process of our scheme is unconditionally secure and non-interactive.*

5.3 Security of the Scheme

In this section, we present security analysis of the proposed scheme. We first consider the security of the sub-share $h^{f_i(j)}$. Given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$, the difficulty of computing the sub-share $h^{f_i(j)}$ is equivalent to breaking the CDH assumption.

Lemma 1. *Under the CDH assumption, it is infeasible to compute the sub-shares from public information.*

Proof. By contradiction, assume that there exists an algorithm $\mathcal{A}$ without knowing $f_i(j)$, which can compute the sub-share $h^{f_i(j)}$ with a non-negligible probability ϵ for the given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. Then, there exists an attacker can solve the CDH problem using the algorithm $\mathcal{A}$. Given $\alpha = g^a, \beta = g^b$ for some $a, b \in \mathbb{Z}_p^*$, we try to compute the value g^{ab} using the capacity of $\mathcal{A}$ in the following.

At random, we pick x, y, z and feed $h = \alpha^x, y_j = h^y, X_{il} = \beta^z, Y_{ij} = h^{yz}$ to $\mathcal{A}$. Since the input to $\mathcal{A}$ is uniformly distributed, we can obtain $h^{f_i(j)} = g^{axbz}$ with

success probability ϵ because of $X_{il} = \beta^z = g^{bz} = g^{f_i(j)}$. By taking $g^{axbz/xz}$, we are thus able to compute g^{ab} with the same success probability ϵ. It is a contradiction to the CDH assumption. □

In the following, we show that participants less than k learn nothing about the secret S. In other words, if no more than $k-1$ participants can recover the secret, it implies breaking the BDH assumption.

Lemma 2. *Under the BDH assumption, it is infeasible that any $k-1$ participants can cooperatively obtain the secret in the proposed scheme.*

Proof. Recalling that the BDH problem is to compute $e(g,g)^{abc}$ for given $g, g^a, g^b, g^c \in \mathbb{G}_1$ where $a, b, c \in \mathbb{Z}_q^*$. A natural variant of the standard BDH problem is to compute $e(g,g)^{aab}$ for given g, g^a, g^b where $a, b \in \mathbb{Z}_q^*$, which is called Computational Bilinear Square (CBS) assumption [18].

By contradiction, assume that there exists an algorithm $\mathcal{A}$ without knowing all $f_i(0) \in A$, which can compute the master secret $s = e(h^{\sum_{p_i \in A} s_i}, h) = e(h^{\sum_{p_i \in A} f_i(0)}, h)$ with a non-negligible probability ϵ for the given public information h, C_{il}, y_j, Y_{ij} such that $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. Equivalently, without knowing some $f_i(0) \in A$, the algorithm $\mathcal{A}$ can compute the $e(h^{f_i(0)}, h)$. Then, there exists an attacker can solve the variant of the BDH problem using the algorithm $\mathcal{A}$.

In the following, we show how to set up the system such that we can compute $e(g,g)^{aab}$. Suppose that participants $P_1, \cdots, P_{k-1}$ are able to break the scheme. At random, we pick some $x, y, x'_j \in \mathbb{Z}_q^*$ and set $h = (g^a)^x, C_{i0} = (g^b)^y, y_j = h^{x'_j}$ for $j = 1, 2, \cdots, n$, which implicitly defines $f_i(0)$ as it required that $C_{i0} = g^{f_i(0)}$. The values $f_i(1), f_i(2), \cdots, f_i(k-1)$ are chosen at random from $\mathbb{Z}_q^*$, which fixes a polynomial $f_i(x)$. This allows us to directly compute $Y_{ij} = y_j^{f_i(j)}$ and $X_{ij} = g^{f_i(j)}$ for $j = 1, 2, \cdots, k-1$. Since $f_i(0)$ is only given implicitly, we cannot compute the values $f_i(k), f_i(k+1), \cdots, f_i(n)$. However, we can use X_{ij} for $j = 1, 2, \cdots, k-1$ to obtain C_{il} for $l = 1, 2, \cdots, k-1$ by solving $k-1$ simultaneous equations $X_{ij} = \prod_{l=0}^{k-1} C_{il}^{j^l}$. When we have computed these values C_{il}, we set $Y_{ij} = (C_{i0} \prod_{l=1}^{k-1} C_{il}^{j^l})^{x'_j}$ such that $Y_{ij} = y_j^{f_i(j)}$, as required for $j = k, k+1, \cdots, n$.

The complete view for the system is now defined. It is consistent with the private view of participants $P_1, \cdots, P_{k-1}$, and comes from the right distribution. Suppose that they are able to compute the master secret $s = e(h^{\sum_{p_i \in A}}, h) = \prod_{p_i \in A} e(h,h)^{f_i(0)}$, then can compute $e(h,h)^{f_i(0)}$. Since we put $h = g^{ax}$ and $C_{i0} = g^{by}$ which implies $f_i(0) = by$, thus we are able to compute $e(h,h)^{f_i(0)} = e(g,g)^{aaxxby}$ with success probability ϵ. By taking $e(g,g)^{aaxxby/xxy}$, we are thus able to compute $e(g,g)^{aab}$ with the same success probability ϵ. It is a contradiction to the variant of the BDH assumption, i.e., CBS assumption. □

By the two lemmas above, we can show that our proposed scheme is secure.

Theorem 5. *Under the CDH assumption and BDH assumption, the proposed scheme is secure, that is, 1) only qualified participants can compute the valid sub-shares; 2) any participants less than k can not recover the master secret.*

As the proof of Lemma 1 and Lemma 2, the correctness of Theorem 5 is straightforward.

6 Performance Analysis

6.1 Computational Complexity

Our scheme consists of three phases: share generation, share verification, and secret reconstruction. For the computation cost, we only consider the "time-consuming computation", which includes modular exponentiation, modular multiplication and pairing operation in each phase.

- Share generation: This phase outputs Y_{ij}, which is encryption of sub-share for each participant, and broadcasts the commitment C_{il}. These requires $n(k+n)$ modular exponentiation .
- Share verification: In this phase, the most time-consuming computation is to verify whether sub-share Y_{ij} is consistent, which needs approximately n^2 pairing operations.
- Secret reconstruction: In this phase, the most time-consuming computation is to verify whether master share s'_j is consistent, which requires k pairing operations.

The summation of operations required in our protocol is $n(k+n)$ modular exponentiation and n^2+k pairing operations.

6.2 Comparison

In this section, we give a comparison of our protocol with those in [7, 12] in terms of computation cost, security of verification, and related properties.

As we analyzed in Introduction, the verifiability of JRSSS [7] is unconditionally secure, however, the scheme can only detect the fact of cheating but not identify who are cheaters. On the contrary, the JRSSS [12] can identify who are cheaters whereas the verification of their scheme is based on the RSA assumption. In addition, the two schemes both do not achieve public verifiability.

In [12], the most time-consuming computation is to verify whether the shares are consistent, which requires $n(n^2+n+k)$ modular exponentiation. In [7], the most time-consuming computation is to reconstruct 100 verification polynomials to verify the strong t-consistency of master shares, which requires $n100k^3$ modular multiplicative.

We denote modular multiplicative, pairing operation, and modular exponentiation by M_m, M_p and M_e, respectively. The comparison of our protocol with those in [7, 12] is shown in Table 1.

We observe that our proposed scheme achieves better properties and stronger security compared with others. Although our efficiency is somewhat lower, but as a compensate for that we first provide JRSSS with public verifiability so far. Furthermore, the verification process in our proposed scheme is unconditionally secure and non-interactive without using the Fiat-Shamir's technique or any additional zero knowledge proof.

7 Application: Dynamic Node-Join in Mobile Ad Hoc Network

In a dynamic topology network, the new nodes need to join or depart it frequently. In this section, using the techniques in our scheme we describe how to add a new node in this environment such as mobile ad hoc network(MANET). In MANET, the secret often acts as a system key and there still exists a same requirement for security, that is, any k nodes or more can recover the system key but nodes less then k lean nothing about it with their shares. After the new node becomes a legitimate member of the MANET, it will possess a share whose format is like others and share the same system key.

There exists a specific dealer to redistribute shares for a new member-join in traditional secret sharing scheme. However, this approach is infeasible in MANET since it is a distributed environment.

If a new node wishes to join the MANET, it must obtain at least k or more nodes approving admission from current MANET and then a new share can be reconstructed cooperatively by k nodes. To maintain the essential security in this process, there are two types of methods. One is to shuffle the secret sharing polynomial by regenerating a random polynomial [9, 21]. The other is to shuffle the partial share by adding blind factor [14, 17]. Nevertheless, these methods lead to a higher computation and communication cost. Since MANET is composed of limited calculation ability, communication capacity and bandwidth, more communications and computation will consume longer time which leads to lower success rate to the generation of new shares.

Herein we employ a more straightforward mechanism to conduct it by combing the techniques in our scheme with Hamiltonian ring instead of the aforemention methods. The detailed is described in Section 7.2.

7.1 Security Requirement

There are security requirement which must be reached in our node-join protocol.

1) Any k nodes or more can recover the system key but nodes less then k learn nothing about it with their shares.
2) Any information about system key cannot be exposed.
3) None but the legal node can get its new share.
4) The shares of old nodes are secure.

7.2 Concrete Protocol

Assume that B is a collaboration of k nodes in MANET, a new node v_r wants to join the MANET. Cooperating parties $v_1, \cdots, v_k \in B$ are arranged in a unclosed Hamiltonian ring for computation the new share s'_r.

1) New node v_r broadcasts/multicasts an joining request among B.
2) Each node $v_j \in B$ calculates a partial share p_j for v_r as follows.

$$p_j = s_j'^{\lambda_j(r)} = h^{\sum_{P_i \in A} f_i(j)\lambda_j(r)},$$

where s'_j is v_j's own master share and $\lambda_j(r) = \sum_{v_i \in B, j \neq i, i=1}^{k} \frac{r-i}{j-i}$. Next, each node v_j masks its private value using v_r's public key $y_r = h^{x_r}$ as follows.

$$\begin{aligned} e(p_j, y_r) &= e(h^{\sum_{P_i \in A} f_i(j)\lambda_j(r)}, h^{x_r}) \\ &= e(h,h)^{x_r \sum_{P_i \in A} f_i(j)\lambda_j(r)}. \end{aligned}$$

3) After v_1 computes $e(p_1, y_r)$, it securely sends it to the next node v_2. Upon receiving $e(p_1, y_r)$, v_2 multiplies it by $e(p_2, y_r)$, to the product $e(p_1, y_r)e(p_2, y_r)$ before sending it to the next node v_3. At the end, last node v_k receives $\prod_{j=1}^{k} e(p_j, y_r)$ and send it to the new node v_r;
4) Using its private key x_r, new node v_r decrypts the last product and obtains master share s'_r as follows.

$$\begin{aligned} s'_r &= \prod_{j=1}^{k} e(p_j, y_r)^{1/x_r} \\ &= e(h^{\sum_{j=1}^{k}(\sum_{P_i \in A} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A}(\sum_{j=1}^{k} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A}(\sum_{j=1}^{k} f_i(j)\lambda_j(r))}, h) \\ &= e(h^{\sum_{P_i \in A} f_i(r)}, h). \end{aligned}$$

Note that the format of new share is different from that of old one slightly. The format is $e(h^{\sum_{P_i \in A} f_i(r)}, h)$ in the former, while that is $h^{\sum_{P_i \in A} f_i(j)}$ in the latter. This does not affect the reconstruction of the system key since there is still a need to calculate $e(h^{\sum_{P_i \in A} f_i(j)}, h)$ before reconstruction in the latter.

Figure 1 shows how this computation operates. Suppose that the communication channels between $v_j \in B$ are secure. For simplicity, we leave out the proof of security in this version.

Table 1: Comparison of three protocols

	[12]	[7]	Our scheme
Computational cost	$(n^3+n^2+nk)M_e$	$(100k^3n)M_m$	$(n^2+k)M_p$
Security of verification	computational	unconditional	unconditional
Public verifiability	no	no	yes
Cheater identification	yes	no	yes
Communication channels	private	private	public

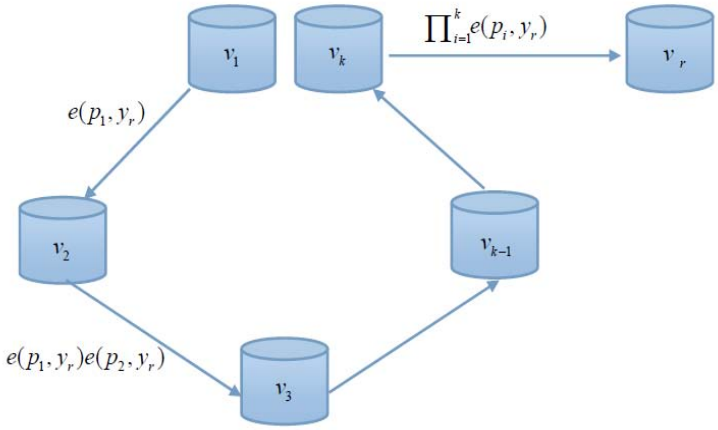


Figure 1: Dynamic Node-Join in MANET

8 Conclusions

In this paper, to the best of our knowledge, for the first time we provide a secret sharing scheme with public verifiability in distributed environment. Utilizing additive homomorphism, each participant acts as a dealer to choose the secret (sub secrets) and generate sub-shares for other participants, which avoids the need for a mutually trusted dealer. By this way, a random master secret will be constructed by some sub secrets corporately. In addition, we explore the technique of homomorphic verification and that of bilinear pairing to allow anyone to publicly verify whether the received shares are consistent. In the verification analysis, we show that the verification process is unconditional secure and non-interactive without using the Fiat-Shamir's technique or any additional zero knowledge proof, which makes it simple and efficient. Finally, we present how our techniques can be applied to handle dynamic node-join in MANET.

Acknowledgment

The authors would like to express their deep appreciation for the valuable comments provided by anonymous reviewers. This work was supported by the National Natural Science Foundation of China (Nos. 61272435, U1261114, 61472146), Guangdong Natural Science Foundation (No. S2013010011859), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2014A030306021), and Research Fund for the Doctoral Program of Xi'an University of Science and Technology (No. 2015QDJ008, 2013QDJ031).

References

[1] C. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret", in *Advances in Cryptology (CRYPTO'86)*, LNCS 263, pp. 251–260, Springer, 1987.

[2] G. Blakley, "Safeguarding cryptographic keys", *Proceedings in AFIPS National Computer Conference*, vo1. 48, pp. 313–317, 1979.

[3] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults", in *Proceedings of 26th IEEE Symposium on Foundation of Computer Science*, pp. 383–395, 1985.

[4] C. K. Chu, S. M. Chow, W. G. Tzeng, et al., "Key-aggregate cryptosystem for scalable data sharing in cloud storage", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 468–477, 2014.

[5] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing", in *Proceedings of 28th IEEE Symposium on Foundations of Computer Science*, pp. 427–437, 1987.

[6] A. Fiat, A. Shamir, "How to prove yourself: Practical solutions to identification and signature prob-

lems", in *Advances in Cryptology (CRYPTO'86)*, LNCS 263, pp. 186–194, Springer, 1986.

[7] L. Harn, C. Lin, "Strong (n, t, n) verifiable secret sharing scheme", *Information Sciences*, vol. 180, no. 16, pp. 3059–3064, 2010.

[8] S. Heidarvand, J. L. Villar, "Public verifiability from Pairings in Secret Sharing Schemes", in *Proceedings of SAC'08*, LNCS 5381, pp. 294–308, Springer, 2009.

[9] A. Herzberg, S. Jaracki, H. Krawczyk, M. Andyung, "Proactive secret sharing or: How to cope with perpetual", in *Advances in Cryptology (CRYPTO'95)*, LNCS 963, pp. 339–352, Springer, 1995.

[10] C. F. Hsu, S. Wu, L. Harn, "New results on ideal multipartite secret sharing and its applications to group communications", *Wireless Personal Communications*, vol. 82, no. 1, pp. 283–292, 2014.

[11] I. Ingemarsson, G. J. Simmons, "A protocol to set up shared secret schemes without the assistance of a mutually trusted party", in *Advances in Cryptology (EUROCRYPT'90)*, LNCS 473, pp. 266–282, Springer, 1991.

[12] K. Kaya, A. A. Selcuk, " A verifiable secret sharing scheme based on the chinese remainder theorem", in *Proceedings of Progress in Cryptology (INDOCRYPT'08)*, pp. 414–425, 2008.

[13] I. Komargodski, M. Naor, E. Yogev, "Secret-sharing for NP", in *Advances in Cryptology (ASIACRYPT'14)*, LNCS 8874, pp. 254–273, Springer, 2014.

[14] J. Kong, P. Zerfos, H. Luo, S. Lu, L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks", in *Proceedings of The Ninth International Conference on Network Protocols*, pp. 251–260, 2001.

[15] P. Kun, "Threshold distributed access control with public verification: A practical application of PVSS", *International Journal of Information Security*, vol. 11, no. 1, pp. 23–31, 2012.

[16] K. Kurosawa, "General error decodable sharing scheme and its application", *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6304–6309, 2011.

[17] L. C. Li, R. S. Liu, "Securing cluster-based ad hoc networks with distributed authorities", *IEEE Transactions on Wireless Communications*, vol. 9, no. 10, pp. 3072–3081, 2010.

[18] B. Libert, D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy reencryption", in *Proceeding sof Public Key Cryptography (PKC'08)*, LNCS 4939, pp. 360–379, Springer, 2008.

[19] Y. X. Liu, L. Harn, C. N. Yang, Y. Q. Zhang, "Efficient (n, t, n) secret sharing schemes", *Journal of Systems and Software*, vol. 85, no. 6, pp. 1325–1332, 2012.

[20] Q. Al Mahmoud, "Polynomial differential-based strong (n, t, n) verifiable secret sharing", *IET Information Security*, vol. 7, no. 4, 312–317, 2013.

[21] X. Y. Meng, Y. M. Li, "A verifiable dynamic threshold key management scheme based on bilinear pairing without a trusted party in mobile ad hoc network", in *Proceedings of IEEE International Conference on Automation and Logistics*, pp. 315–320, 2012.

[22] T. P. Pederson, "Non-interactive and information-theotetic secure verifiable secret sharing", in *Advances in Cryptology (CRYPTO'91)*, LNCS 576, pp. 129–140, Springer, 1991.

[23] T. P. Pedersen, "A threshold cryptosystem without a trusted party", in *Advances in Cryptology (EUROCRYPT'91)*, LNCS 547, pp. 522–526, Springer, 1991.

[24] A. Shamir, "How to share a secret", *Communications of ACM*, vol. 33, no. 3, pp. 612–613, 1979.

[25] M. Stadler, "Publicly verifiable secret sharing", in *Advances in Cryptology (EUROCRYPT'96)*, LNCS 1070, pp. 190–199, Springer, 1996.

Zhenhua Chen received her Ph. D. degree from Shaanxi Normal University in 2014. She is currently a associate professor at School of Computer Science and Technology, Xi'an University of Science and Technology. Her research interests include secure multi-party computation and secret sharing.

Shundong Li received his Ph. D. degree from Xi'an Jiaotong University in 2003, and had been studied in Tsinghua University as Postdoctor from 2003 to 2005. From 2005 to 2007, he had been a associate professor at Beijing Normal University. He is currently a professor and supervisor of Ph.D. at Shaanxi Normal University. His research interests focus on secure multi-party computation and confidential data mining.

Qiong Huang received his Ph.D. degree from City University of Hong Kong in 2010. He is now a professor with College of Informatics, South China Agricultural University. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis.

Jianhua Yan is now a PH.D. candidate of Beijing university of posts and telecommunications. His research interests focus on lattice-based cryptography and the security of cloud.

Yong Ding received his Ph. D. degree from Xidian University in 2005, and had been studied in City University of Hongkong as Research Fellow from 2008 to 2009. He is currently a professor and assistant dean at School of Mathematic and Computing Science, Guilin University of Electronic Technology. His research interests focus on cryptography and information security.

Trust Based HWMP Protocol in High-Performance Wireless Mesh Networks

Parimalla Subhash[1] and S. Ramachandram[2]
(Corresponding author: Parimalla Subhash)
Department of CSE, Jyothishmathi Institute of Technological Sciences[1]
Karimnagar, India
(Email: subhash.parimalla@gmail.com)
Department of CSE, UCE, Osmania University[2]
Hyderabad, India
(Received Sept. 6, 2015; revised and accepted Nov. 12, 2015)

Abstract

Wireless mesh networks are multi hop wireless networks with high performance requirements. To enhance the performance, a large number of routing protocols have been proposed focussing on various link properties.The metrics designed to capture various link properties make an important assumption that nodes cooperate in network operations. On the other hand, the nodes are spread over larger area and maintained by different operators which lack cooperation leading to selfish and malicious behavior. To address this issue, several works have been carried out by modelling trust/reputation into a network. In this paper, we modify an existing trust based secure routing framework, AODV-REX, tailored to mesh networks, as a first step. We observe that the existing trust models for distributed wireless networks are not directly employable and need to be significantly modified to meet the performance requirements of mesh networks. Further, we propose a trust extension to HWMP (Hybrid Wireless Mesh Protocol), called HWMP-TX based on a new trust model. The analysis and simulation results show that HWMP-TX is resilient to various internal attacks and achieves better performance.

Keywords: HWMP, reputation, secure routing, trust, wireless mesh network

1 Introduction

Wireless Mesh Network (WMN) is a multi-hop wireless network that inherits self-healing, and self configuring capabilities from mobile ad hoc network (MANET). Besides these, the additional features of WMN include static, and non-power-constrained nature of mesh routers. WMN also lowers the deployment cost and administrative overhead by replacing the majority of the wired infrastructure. These features make them an ideal candidate solution for providing wireless broadband internet access in an office, campus or community networks, without requiring every access point to be physically connected to the Internet [2]. Thus, WMN technology has generated a huge amount of interest in the industry and academic fields due to its suitability to various commercial application scenarios.

On the other hand, there are many other issues that need to be addressed to make WMNs commercially successful. Design of an optimal routing protocol is one such issue that requires immediate attention. As WMNs are expected to support high performance internet applications, routing protocol and the employed routing metric plays a dominant role in determining the amount of throughput achieved. Several routing protocols have been proposed in conjunction with different routing metrics to increase the overall performance of the network [5, 6, 11, 25].

Design of routing metrics for WMN mainly involves accounting for the physical properties of a wireless link that usually affect the network performance. It should also account for the features that indirectly contribute to the network performance. Properties, mainly, link variability, varying available bandwidth and flow interference (inter and intra) should be considered to maximize throughput [22]. The design of routing metric that results in optimal performance assume that all nodes are honest and well behaved in the route selection process. This is not a valid assumption in a distributed network like WMN, where nodes operate in an open wireless environment. Nodes tend to exhibit selfish and malicious behavior, and needs to be accounted to enhance performance.

Routing misbehavior is one major issue in any distributed network like WMN. The existence of selfish nodes is justified due to the presence of nodes from multiple operators in a commercial WMN. These nodes may intentionally drop the packets, forwarding their own traffic. Nodes can also be easily compromised by an adversary,

due to the open environment in which they operate. To overcome these kinds of routing mis-behavior, variety of protocols have been proposed particularly by employing trust or reputation in routing activity [1, 8, 12, 14, 17, 18, 24]. The way these protocols employ trust in the routing process depends on the network requirements. For example, in a MANET, where the focus is on maintaining end-to-end connectivity, protocols directly employ trust to select relatively higher trustworthy paths. In a WMN, trust modeling is more complicated because of their need to support high performance applications.

Routes established in a WMN should meet the throughput requirements as well as the trust criterion of the network. In such a scenario, employing trust as the sole metric in route selection process may meet trust requirements, but fail to achieve desired throughput, as the employed metric ignores wireless link characteristics. Integrating trust value of a node/link with the underlying routing metric is an alternate way of discovering routes [17, 18]. But, this integration scheme also does not achieve good results and these two entities (routing metric and trust) are independent and if integrated fail to achieve optimal performance in certain cases.

In this paper, we observe that the existing trust models for distributed wireless networks are not directly employable and need to be significantly modified to meet the performance requirements of mesh networks. As a first step, we modify an existing trust based routing framework, AODV-REX, tailored towards mesh network. Further we propose a trust extension to HWMP, called HWMP-TX based on a new trust model. It complements the link-quality-based routing metric in making routing decisions and achieves better performance over existing approaches to employ trust. We specifically focus on HWMP along with airtime link metric, as it is the mandatory routing protocol to be implemented, according to IEEE 802.11s draft standard for 802.11 based mesh networks [15]. The analysis and simulation results show that HWMP-TX is resilient to various internal attacks and achieves better performance. The performance of both the models is evaluated under various attack scenarios.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Internal attacks against HWMP are presented in Section 3. In Section 4, we propose our modifications to AODV-REX framework that integrates the reputation metric with high throughput path selection metric like airtime. Later, in Section 5, we propose a complete trust model based on an alternate mechanism to employ trust. Performance evaluation and security analysis is presented in Section 6. Experimental results in comparison with existing model are discussed in Section 7. Discussion of several factors is included in Section 8. Finally, Section 9 concludes the paper.

2 Related Work

Lately, a lot of research has been carried out to increase the performance of routing protocols for WMN.The main design goal of these protocols is throughput maximization. Numerous link-quality-based routing metrics have been proposed replacing hop count, to increase the overall throughput, as it has been shown that the hop count selects sub-optimal routes [5]. Metrics such as ATLM (airtime link metric) [16], ETX (expected transmission count) [5], ETT (expected transmission time) [6], WCETT (weighted cumulative ETT) [6] and mETX (modified ETX) [25] have been developed replacing hop count. The main design aim is to enhance performance and increase throughput. The existing reactive and proactive routing frameworks are modified accordingly to accommodate the designed metrics. For example, multi-radio link quality state routing protocol (MR-LQSR) is based on optimal link state routing protocol (OLSR), enhanced to accommodate multiple radios and WCETT routing metric. These metrics are modelled by assuming the co-operation among participating nodes. This is an optimistic assumption in a distributed network like WMN where nodes operate in an open environment, and the possibility of nodes being compromised by an adversary cannot be ignored.

The problem of routing security and node misbehavior has been studied by different researchers, e.g., [1, 8, 12, 14, 17, 18, 24]. Various trust based routing protocols have been proposed for ad hoc and WMNs to mitigate the influence of these malicious nodes in the route selection process.

The distributed trust model proposed by Rehman et al. [1] assumes discrete levels of trust. It employs a decentralized approach to manage trust and a recommendation protocol to exchange trust related information. The model is based on a conditional transitive trust relation that uses trust categories to express trust towards other agents. In order to establish trust relation between entities where a direct relation does not exist, the agents can make use of an intermediate agent to establish trust. Various trust models that exist in the literature try to quantify the trust relationships according to different applications security requirements.For example, the PGP style authentication schemes with certification chains use binary trust valuation

Zheng et al. [24] proposed a trust model that assigns quantitative trust value to each node based on the observed behavior. A node evaluates its relationship with other nodes in a network, based on factors such as experience statistics (es), data value (d), intrusion detection result called intrusion black list (ibl) and references (r) along with a node's preference and policy. Each node maintains a trust matrix to store the knowledge accumulated on the above factors for every other node in the network with the help of network traffic monitoring and recommendations. That is, every node maintains trust relations with all other nodes in the network. Final trust

evaluation of node i to node j for an action a is evaluated through a linear equation that uses the values stored in the trust matrix. The evaluated trust values are used for making better routing decisions. As, each node maintains a list of values for various factors for every other node in the network, the overhead in decision making is very high.

TAODV proposed in [8] is a trusted extension to AODV. The path selection process is similar to AODV with trust replacing hop count as the routing metric. The trust values of nodes are assumed to be distributed in prior. Hence, it does not model the way trust relations are established and fostered. To incorporate trust into the route selection process, the route request ($RREQ$) header is modified to include a trust level field in the AODV RREQ. When a node receives a RREQ, it rebroadcasts it after modifying the trust level field with the trust value of the node from which it received the RREQ. Every node checks back the rebroadcasted RREQ from its previous node to see whether it has provided the proper information. If not, it sends a route warning message questioning the sanctity of the node. The final route selection is based on trust level metric. The major drawback of this model is the prior distribution of the trust levels. Moreover, there is no mechanism to modify the established trust-levels depending on the change in nodes' behavior.

Eissa et al. [7] proposed FrAODV,a friendship based AODV protocol to establish secure paths. It is similar to that when a person (X) wants to verify another person (Y),he generally asks his friends about this person. He also asks this person to provide him with the list of reference persons,who will be asked if he is to be trusted. This protocol uses two algorithms i.e FwEvaluate algorithm to evaluate the forward routes and the RvEvaluate algorithm to evaluate reverse routes in AODV protocol.

Meka et al. [14] proposed a trust framework for AODV that employs trust as the routing metric instead of hop count. According to this framework, a node maintains trust relationships with its neighbors. It also allows a node to assign trust levels to the routes that it discovers. Each node maintains a neighbor trust table (NTT) to store the neighbor ID, its trust value and the current number of RREQ's it can send. In addition to maintaining the NTT, the routing table is modified to include all the routes from that node to a destination, to incorporate route trust. Each node keeps track of the number of packets it has forwarded through a particular route. Trust relations are evaluated with the help of a route acknowledgement RACK that a destination node periodically sends addressed to the source, which contains the number of packets received till that time instant. All the intermediate nodes along the reverse route make use of the RACK to compute the route trust. Whenever a node generates or forwards a RREP, it advertises its trust value (ATV) on the route under consideration to its immediate upstream node. Based on the ATV and the observed trust value (OTV), a node updates the node trust for that neighbor.

Two-Hop acknowledged routing protocol (2-HARP) proposed in [26] is based on zone routing protocol.In 2-HARP, each node maintains trust relations with all the nodes in its 2-hop neighborhood using the neighbor sensing mechanism of OLSR [4]. Each node maintains an acknowledgement table in addition to the routing table. The acknowledgement table is used to store information about packets waiting to be acknowledged. A node after sending a packet, expects a signed acknowledgement from the 2-hop neighbor on the established route, to verify whether the one hop neighbor on the established route, has indeed forwarded the packet. If the one hop neighbor intentionally drops a 2-hop acknowledgement, the 2-hop neighbor tries for a maximum of s times before labelling the node as non-responsive. The main drawback of this model is the acknowledgement overhead. As, each data and control packet is acknowledged twice, it incurs very high overhead.

Tan et al. [23] proposed a trust reasoning model based on fuzzy Petri net is presented for the evaluation of trust values of mobile nodes. In addition, to avoid compromised or malicious nodes, a trust based routing mechanism is proposed to select a path with the highest path trust value among all available paths. Further, OLSR is extended by using the proposed trust model and trust based routing mechanism, called FPNT-OLSR. For the implementation of FPNT-OLSR, a trust factor collecting method and trust information propagating method is designed, which do not generate extra control messages.

AODV-REX proposed in [17] is a reputation based extension to AODV for WMNs. Acoording to AODV-REX, a node maintains two different kinds of reputation values for each of its neighbors (local and global).Local reputation of a neighbor is based on nodes' direct observations using a watchdog [12]. Global reputation of a node is computed based on reputation values obtained from other nodes in the network. Whenever a node requests for a route towards a destination, it transmits a RREQ by appending the reputation values and addresses of all its neighbors. An intermediate node that receives a broadcasted RREQ, acts on the reputation values of interest in the RREQ and ignores the rest leaving them unmodified. It re-broadcasts the RREQ by further appending it with the reputation values of its neighbors. The hop-count metric is modified to accommodate the reputation of a node. The basic idea is to create a new virtual distance that takes into account the reputation level of the node connected to the link. The distance between two neighboring nodes increases if the reputation of one of the node decreases and so the route will be less considered. AODV-REX incurs huge routing overhead as each intermediate node appends the RREQ with the trust values of all its neighbors, thus increasing its size enormously. It is also based on hop count metric that has been shown to select sub-optimal routes

EFW (expected forwarding counter) proposed in [18] is a cross-layer metric for routing in WMN that considers malicious and selfish participants. It employs watchdog to monitor the forwarding behavior of its neighbors. The

forwarding ratio of a node is integrated with ETX (estimated transmission count of a link) to derive a cross-layer routing metric called as EFW. To summarize, for calculating EFW of a link a node needs to monitor its neighbors, calculate its forwarding ratio, and integrate that value with the existing ETX metric.

A key point to note is that in all of the above existing work, trust is either directly employed or integrated with the employed routing metric. Even the attempts to integrate trust with the routing metric have been made on hop count except EFW, where the forwarding probability of a node is integrated with high throughput metric, ETX. In Section 5 , we present an alternate mechanism to employ trust in the routing process that is shown to perform better on average than the attempts to integrate with the routing metric.

Moreover, the majority of the frameworks discussed above employ watchdog to evaluate trust relations, which restricts the nodes from efficiently using the available resources thus affecting network performance.

3 Internal Attacks on HWMP

In this section, we focus on various possible attacks on HWMP (Hybrid Wireless Mesh Protocol). We specifically focus on HWMP as it is the mandatory routing protocol for IEEE 802.11s based mesh networks. We specially concentrate internal attacks, as the authentication protocol at the MAC layer acts as a first layer of defense against attacks from external nodes. Before discussing the various internal attacks, a brief overview of HWMP is provided to understand the operation of the protocol.

3.1 Overview of HWMP

HWMP is a hybrid wireless mesh protocol [3] that operates at layer-2 and employs MAC addresses for path selection. It is called a hybrid protocol as it combines both reactive and pro-active routing strategies. It combines the flexibility of on-demand route selection with proactive topology tree extensions. The combination of reactive and proactive elements of HWMP enables efficient path selection for a wide variety of mesh networks. HWMP is based on ad hoc on-demand distance vector (AODV) protocol adapted for MAC-address based path-selection and link metric awareness [19].

HWMP provides two modes of operation, they are on-demand mode and proactive mode. These two modes are not exclusive and are used concurrently, because the proactive modes are extensions of the on-demand mode. HWMP uses four different kinds of information elements (IEs). Path Request (PREQ), Path Reply (PREP) and Root Announcement (RANN) are employed in path-selection process, while Path Error (PERR) is used for route-maintenance. In HWMP, a path to the destination is described by the next hop at every intermediate mesh station (mesh STA). When a source wants to send data to a destination for which it does not have a path yet, it initiates a path discovery by broadcasting a PREQ.The PREQ Information Element is shown in Figure 1 contains various fields out of which Hop Count, Element TTL, Metric and the Target-Only sub field in Per Target field are operated upon by intermediate nodes as part of the path selection process.

The hop-count field is acted upon by every intermediate node along the selected path. Its value is set to an integer equal to the number of hops from the originator STA to the mesh STA transmitting the PREQ.Element TTL field indicates the remaining number of hops allowed for the PREQ element. It is mainly used to prevent the PREQ element from traversing the network endlessly. Initially, the value of TTL element is set to a number equal to the network diameter. The metric field is set to the cumulative metric from the originator to the mesh STA transmitting the PREQ. The IEEE 802.11s specifies the use of air-time link metric (ATLM) as the default link metric to identify an efficient radio-aware path. All the above discussed fields are modified by the intermediate nodes accordingly, enabling better path selection.

3.2 Attacks on HWMP

HWMP is prone to a number of security attacks from internal malicious nodes. A compromised node becomes an epicentre for launching a variety of attacks, thus degrading the network performance rapidly. Attacks are usually aimed at disrupting the normal network operations. The various kinds of internal attacks are discussed below.

3.2.1 Flooding

It is one of the most simplest and efficient attack. In HWMP, a node can generate any number of PREQs. Malicious nodes can exploit this and flood the network with a number of PREQ's for non existing nodes. A legitimate node would be forced to spend majority of its time processing the PREQ's, resulting in severe performance degradation [20]. Such an attack can be countered by limiting the number of PREQ's that a node can generate based on its trust level.

3.2.2 Modification Attacks

Malicious nodes can redirect network traffic, and launch DoS attacks by altering the fields in IEs. For example, a malicious node can modify the metric field in the PREQ element to include itself in the selected path. Once included, it can launch various other packet dropping attacks. Such attacks can be specifically called as metric manipulation attacks. As, nodes need to cooperate in determining the metric of a path, malicious nodes can exploit this to their advantage. The sequence number of a PREQ message can be modified by a malicious node to a value much higher than that of the destination's current sequence number, thereby fooling the originator of

Element ID	Length	Flags	Hop Count	Element TTL	PREQ ID	Org.Mesh STA.Addr.	Org. HWMP Seq.NO
Org.Ext Addr	Life Time	Metric	Target Count	per Target addr #1	Target Address #1	Target HWMP Seq.No	.....

Figure 1: PREQ element

the PREQ to believe that the manipulated PREP as genuine. There is no way to distinguish between path replies generated by an intermediate node and the destination node, therefore this attack has significant impact on the selection of network paths.

3.2.3 Wormhole Attacks

Wormhole is a hypothetical channel formed between two colluding nodes. The main aim of this attack is to disrupt the routing functionality of a network. A Wormhole can be created by simply tunnelling messages between two colluding nodes, or by transmitting them on an out-of-band channel or by just relaying packets [10, 13, 21]. Once a wormhole is established, the two colluded nodes at the either ends of the tunnel can use this channel to influence path selection decisions. A malicious node can tunnel a PREQ through an out-of-band channel and replay it at the other end. As a path formed through these colluded nodes inherently offers better metric over other available paths. Once, a path is established, the colluded nodes can launch various packet dropping attacks.

3.2.4 Fabrication Attacks

A malicious node can fabricate messages to disrupt the network operations. For instance, a malicious node can fabricate a PERR message that is actually used to notify the nodes along the downstream that the next hop to the originator of PERR is currently unavailable. Nodes receiving such a message will mark the link as broken and re-initiate path discovery. As, cryptographic solutions cannot prevent such kind of internal malicious attacks, an efficient detection mechanism is required to detect and exclude the malicious node. Employing trust to detect such malicious behavior has attracted much research attention and several trust frameworks have been developed to address this issue. Even though several trust frameworks exist in literature, they cannot be directly employed due to the high performance requirements of WMN. Therefore, there is a need for a framework that concurrently focuses on performance requirements on the one hand and trust requirements on the other.

4 Modified AODV-REX for Wiress Mesh Networks

HWMP, the routing protocol for WMN is based on AODV and AODV-REX is a reputation based extension to AODV that integrates reputation of a node with the hop count. As, hop count has been shown to select sub-optimal paths, we attempt to modify AODV-REX for WMN by integrating reputation of a node with airtime metric (ATLM) [3]. We refer to this modified AODV-REX as HWMP-REX. The airtime link metric is a measure for the amount of the consumed channel resources when transmitting a frame over a particular wireless link. The following Equation (1) is used to calculate airtime metric of each link.

$$C_a = \left[O_{ca} + O_p + \frac{B_t}{r}\right] \frac{1}{1 - e_{pt}}. \quad (1)$$

The airtime cost for each pair wise link C_a is calculated in terms of the modulation rate (r) and bit error rate e_{pt} for a test frame of B_t size. Where, O_{ca} is the channel access overhead, O_p is the protocol overhead. O_{ca}, O_p and B_t are constants defined for each 802.11 modulation type.

4.1 HWMP-REX

The proposed modifications are primarily concerned with integrating reputation of a node with airtime rather than hop count. The reputation model and the reputation dissemination process of AODV-REX are left unmodified. AODV-REX employs a multi-layered model for estimating the reputation of network nodes, called REFACING (RElationship-FAmiliarity-Confidence-INteGrity) [16]. It maintains two kinds of reputation values for its neighbors-local and global. Local reputation of a neighbor is based on node's direct observations using a watchdog [12]. Global reputation of a neighbor is computed from reputation values provided by other nodes in the network.

As described in Table 1, when a node has data to send, it generates a PREQ message. Together with the usual HWMP information, a node appends the reputation and addresses of its neighbors to the PREQ message. Upon reception of such PREQ, a node acts on the reputation values of interest and ignores the rest leaving them unmodified. In addition to the reputation values, a node

also acts on the metric field of the PREQ message, as part of the normal process of processing a PREQ. The metric computation process is modified to include the reputation of a node from which it received the PREQ. The modified reputation metric (RM) of a link from node A to node B is given by Equation (2).

$$RM\left(\overrightarrow{AB}\right) = [(1 - R_{B_A}) * AD] \quad (2)$$

R_{B_A} is the reputation of a node B in A and AD denotes the Airtime Diameter. Airtime Diameter is the airtime taken for a standard frame to traverse between two ends of the network. AD can be computed with the help of a test frame, transmitted by setting the TTL equal to the network diameter, at the time of network initialization. The reputation metric of a link is added to the airtime of a link to get the resultant metric. As, the reputation of a node decreases, the reputation metric of a link increases, increasing the overall airtime thus avoiding malicious nodes.

4.2 Issues in HWMP-REX

In HWMP-REX, one of the major issue is computing airtime diameter, AD, of a network. Determining AD is a complex task as it has to be calculated after the network has been initialized. Whenever new nodes are added, the airtime diameter needs to be re-computed for determining overall routing metric of a path. The other important issue is high fluctuations in path selection. This behavior is due to the fact that, HWMP-REX selects a path $\overrightarrow{P_o}$ based on the cumulative metric obtained by integrating reputation of a node with airtime metric of a link. This can be represented using Equation (3), where $l_{s-j(airtime)}$ gives the airtime of a link l_{s-j} and $RV_{l_{s-j}}$ gives the reputation of a node S in J associated with the link.

$$\begin{aligned} RM\left(\overrightarrow{P_o}\right) &= \sum_{L_{\overrightarrow{P_o}}} \left(l_{s-j(airtime)} \oplus RV_{l_{s-j}}\right) \\ &= \left(l_{s-j(airtime)} \oplus RV_{l_{s-j}}\right) \\ &\quad + \left(l_{j-k(airtime)} \oplus RV_{l_{j-k}}\right) + ... \\ &\quad + \left(l_{n-d(airtime)} \oplus RV_{l_{n-d}}\right) \end{aligned} \quad (3)$$

As, both components of Equation (3) play an equal role in path selection process, falsely penalizing genuine nodes result in path fluctuations. As, no lower bounds are established for HWMP-REX to distinguish malicious behavior from normal behavior, it naturally prefers nodes with high reputation over nodes with lesser reputation. In such cases, HWMP-REX prefers a path with better overall cumulative metric over a path that actually achieves higher throughput, which indirectly is a false positive. Existence of lower bounds allows the system to differentiate malicious behavior from normal, which in turn allows the network to chose a lesser reputation node due to its higher link quality. These bounds cannot be established for HWMP-REX, as the path selection decision is based on integrated metric that strictly prefers high reputation nodes over nodes with relatively lesser reputation. To overcome these limitations, we propose a new trust based routing approach that is based on an alternate mechanism to employ trust in routing process.

5 The Proposed Secure Routing for WMN

The proposed secure routing scheme for WMN is based on a new trust model that employs a different approach to employ trust in the route selection process and defends internal attacks.

5.1 Proposed Trust Model

The proposed trust model complements HWMP with its trust observations and allows it to select high throughput trustworthy paths without integrating with the airtime link metric. The trust model comprises of three different phases that are carried out independently without intervening with the routing process. Those are Initialization, Trust Evaluation and Trust Recommendation. The various symbols used in this paper are given in Table 2.

Table 2: Symbols used and their meaning

Symbol	Meaning
U_{ij}	Initial Trust Value
V_{ij}	Current Trust Value
β_{ji}	Packets received by i from j during time interval $TE_{interval}$
$TE_{Interval}$	Trust Evaluation Interval
ε_l	Expected Loss in the Network
δ	Small Fractional Value
Υ_l	Lower Threshold Value
Υ_u	Upper Threshold Value
τ_l	Tolerance Level
ψ_{ij}	Revised Trust Value of j in i

5.1.1 Initialization

A node I after discovering its set of neighbors $\{J\}$, initializes them to a trust value (U_{ij}) of 0.5. The value 0.5 is justified as a node neither trusts nor distrusts the neighbor. The maximum trust value that a node can attain is 1.

5.1.2 Trust Evaluation

Each node periodically evaluates the behavior of its neighbors using the trust evaluation procedure given in Algorithm 1. The evaluation procedure is carried out independently by each node and the evaluation timing of nodes

Table 1: HWMP-REX path selection process

Executed at the source node S initiating Path Discovery process
1: Create a PREQ element by appending the reputation and addresses of all the neighbors of S. **2:** Set the Metric field to 0. **3:** Broadcast the PREQ.
Executed at the intermediate node J upon receipt of the PREQ
1: Parse the PREQ element to act on the reputation fields with which J shares neighborhood. **2:** Update the global reputation of those nodes. **3:** Append the PREQ element with the reputation values and addresses of Js neighbors. **4:** Update current link metric: Metric = currentMetric + (RM + airtime) **5:** ***if***(*Route to source is available*) *then* Unicast PREP Rebroadcast PREQ **6:** ***else*** Rebroadcast PREQ
Executed at the Destination node D upon receipt of PREQ
1: Parse the PREQ element to act on the reputation fields with which D shares neighborhood. **2:** Update the global reputation of those nodes. **3:** Include current link metric: Metric = currentMetric+(RM+airtime) **4:** Choose a path with the best metric. **5:** Unicast the PREP.

need not be synchronized. The evaluation of a neighboring node's behavior is based on the assumption that all the nodes in the network are fairly loaded. This assumption is justified in a WMN as wireless mesh routers are dedicated routers that provide continuous access services to its clients when they are in operational mode. Hence, the contribution of every genuine node in forwarding the network traffic is approximately equal. According to the trust model, a node monitors the performance of its neighbors during an interval of time denoted by $TE_{interval}$. During this time interval $TE_{interval}$, an evaluator node I expects a fixed number of packets α_{ji} from each of its neighboring nodes J periodically. A node also considers the transient losses in the network due to congestion, collisions and errors in the network channel denoted by ε_l .

At the end of the time interval $TE_{interval}$, node I computes the difference between number of packets actually received (β_{ji}) from neighbor J to the packets estimated. After accommodating network losses, if β_{ji} does not confer with estimate α_{ji} , then the additional drop in packets is considered to be an intentional and J is penalized by decreasing its trust value by δ for each packet dropped. A tolerance level of τ_l is allowed to accommodate dynamic variation in channel conditions. If the trust value of node J falls below a threshold value Υ_u(upper-threshold), then I requests for a recommendation about that particular neighbor J. The evaluation time interval can be set accordingly, i.e. the duration can be short or long depending on the type of application in which the model is employed.

The upper and lower threshold values are just to facilitate the characterization of malicious activity of nodes. A higher lower threshold allows protocol to converge quickly, thus identifying malicious behavior. This may sometimes results into falsely ignoring genuine nodes . For a higher percentage of malicious nodes , HWMP-TX incurs higher losses , as it takes more time to converge. This behavior can be attributed to the optimistic nature of HWMP-TX protocol.

Algorithm 1 Trust evaluation

```
1: Carried out by each node I at the end of their TE_interval
2: for each neighbor J do
3:    if (β_ji > α_ji − (ε_l + τ_l)) then
4:       V_ij = u_ij + δ //good behavior
5:    else if (β_ji < α_ji − (ε_l + τ_l)) then
6:       V_ij = u_ij − δ //Suspicious behavior
7:       if (Vij < Υ_u) then
8:          requestRecommendation(J )
9:       end if
10:   else if (β_ji == α_ji − (ε_l + τ_l)) then
11:      V_ij = u_ij //expected behavior
12:   end if
13: end for
```

5.1.3 Trust Recommendation

Trust recommendation procedure shown in Algorithm 2, is reactive one carried out by a node I when the trust value of a neighbor J falls below Υ_u .

Algorithm 2 Trust recommendation

1: Executed by node I after receiving r recommendations
2: $\psi_{ij} = \frac{T_{ki}*T_{kj}+T_{li}*T_{lj}+....+T_{ri}*T_{rj}}{r}$
3: **if** $\psi_{ij} < \Upsilon_l$ **then**
4: M = True // Node I sets J status to malicious
5: **else if** $\psi_{ij} > \Upsilon_u$ **then**
6: Continue normal network operations
7: **else if** $\Upsilon_l < \psi_{ij} < \Upsilon_u$ **then**
8: Closely monitor J
9: **end if**

Nodes that receive a request for trust recommendation, check their respective neighbor list to verify the existence of J. If J exists in their neighbor list, it replies to the request sent by I with the current trust value of J in its list. Once, node I receives all the recommendations, it re-evaluates the trust value of J. If the revised trust value denoted by ψ_{ij}, falls below Υ_l, then node I assumes J to be malicious.

5.2 Trust Based Secure Routing (HWMP-TX)

The proposed trust model works in conjunction with HWMP to enable better route discovery process. It periodically evaluates the behavior of each of its neighbors by monitoring their forwarding behavior. It allows the routing protocol HWMP, to establish secure end-to-end routes by providing it with the observed trust values. The path selection process of HWMP-TX is shown in Table 3.

A source node O initiates a route discovery process by broadcasting a PREQ for a destination node D. An intermediate node I that receives a broadcasted PREQ, first verifies whether the trust value of the transmitter (For example, O in the first turn) is above a predefined threshold Υ_u. If the transmitter does not meet the desired trust requirements, PREQ's from such nodes are not processed further. This process is repeated by each intermediate node until the PREQ reaches destination or a node that has fairly fresh route to the destination. Finally, when the PREQ reaches the destination D, it too verifies the trust value of the transmitter, and selects a better route, before unicasting a PREP. The trust model ensures that the nodes included in the path, pass the basic trust acceptance criteria. Overall the route selection process is mainly driven by the airtime of a link and the trust model complements the path formation by ensuring that the selected nodes satisfy the basic acceptance criteria.

6 Security Analysis and Performance Evaluation

6.1 Security Analysis

6.1.1 Flooding Attack

An internal malicious node can generate any number of PREQ's requesting for paths to non-existent destinations. This attack can be easily handled by limiting the number of requests that a node can generate depending upon their reputation. HWMP-TX can naturally handle this kind of attack as the requests from a node are processed if and only if they satisfy the minimum trust criterion. As, HWMP-REX does not establish any lower bound on the reputation levels, a node can flood the network with fake requests.

6.1.2 Metric Manipulation Attack

The most common modification attack in a high throughput network is a metric manipulation attack. Malicious nodes can manipulate the metric field to include themselves in the selected path and later launch various packet dropping attacks. These attacks can be successfully launched before the nodes begin their monitoring process. Once, trust relationships are established, and nodes begin their monitoring process, the success percentage of these attacks falls drastically. In HWMP-REX and HWMP-TX, as nodes with lower reputation are avoided in path selection process, this kind of attack can be usually detected over time.

6.1.3 Wormhole Attack

Malicious can act in collusion to record packets at one end of the network and replay them at the other end. The main aim of this attack is to convince two far away nodes as neighbors. Once, a wormhole is established, the nodes can launch several packet dropping attacks. HWMP-REX fails to detect attacks from such colluded nodes as the watchdog does not guarantee reception at the receiver. HWMP-TX can identify such colluded nodes independently with the help of other nodes sharing neighborhood with such a malicious node and it can detect packet dropping attacks.

6.1.4 Blackhole Attack

Any malicious node's ultimate goal is to disrupt network services, and blackhole attack is the easiest and most straight forward among all the attacks. For a node to behave as a blackhole, first it has to become a part of the selected path. Hence, a blackhole attack is always launched in conjunction with other attacks such as, a metric manipulation attack. A node cannot behave as a blackhole for an extended period of time as the trust evaluation mechanisms of both HWMP-TX and HWMP-

Table 3: HWMP-TX path selection process

Executed at the source node *S* initiating Path Discovery process
1: Create a PREQ element similar to HWMP
2: Set the Metric field to 0
3: Broadcast the PREQ
Executed at the intermediate node J upon receipt of the PREQ
1: Check the PREQ-ID to avoid processing duplicate PREQ's
2: ***if*** $(duplicate)$ *then* Drop the PREQ return; ***else*** verify the trust value of the transmitter. // S in the first run
3: ***if*** $(TV_{Transmitter} < \Upsilon_u)$ *then* Drop the PREQ ***else*** Process the PREQ similar to HWMP
4: Update current link metric: Metric=currentLinkMetric + Metric.
6: ***if*** $(Route\ to\ source\ is\ available)$ *then* Unicast PREP Rebroadcast PREQ ***else*** Rebroadcast PREQ
Executed at the Destination node D upon receipt of PREQ
1: Verify the trust value of the transmitter to satisfy acceptance criteria
2: Include current link metric: Metric = currentMetric + Metric
3: Choose a path with the best metric.
4: Unicast the PREP.

REX can easily detect such attacks and avoid such nodes in future path selection process.

6.1.5 Fabrication Attack

Selfish nodes can fabricate messages to avoid consumption of their resources. For example, a node can fabricate a PERR message to inform the downstream nodes about an active link as broken. HWMP-REX inherently cannot handle such an attack as there is no way to distinguish between a genuine and fabricated PERR message. In HWMP-TX, for a node to maintain neighbor relations, it needs to keep its links active and this restricts a node from frequently generating fabricated messages.

7 Experimental Results

7.1 Simulation of Trust

We evaluate the performance of HWMP-REX and HWMP-TX by performing the following simulations on Omnet++ 4.2.1 discrete event simulator. We consider a backbone mesh network of 40 uniformly distributed mesh routers placed over the area of 2000 m^2. The transmission range of each mesh router is set to 100m. Each simulation is performed 10 times and the average results are presented. Mesh routers implement 802.11s MAC protocol with a channel data rate of 11mbps. Nodes uses CBR data traffic.

To begin with, we analyze the throughput achieved by HWMP-REX and HWMP-TX. Out of 40 mesh routers 20% of mesh routers (i.e. 8 nodes) exhibit malicious behavior. Malicious nodes are strategically selected in such a way that they become part of the network path. Figure 2 summarizes the performance of both the protocols. HWMP-REX achieves higher throughput during the initial stages of network activity. However as the simulation time progresses the throughput gradually falls below HWMP-TX. On the other hand, HWMP-TX achieves higher throughput as the life time of the network progresses . This is due to the fact that HWMP-REX assigns higher weighage to the resulting trust values in comparison to HWMP-TX. Since, any kind of malicious activity lowers the trust, thus increasing the virtual distance there by allowing HWMP-REX to select higher trusted paths over high throughput paths. But, HWMP-TX waits for an interval of time (When trust falls below higher threshold) before ignoring malicious nodes. HWMP-TX employs only airtime metric to select routes and trust is used to check whether the nodes meet the minimum trust criterion. Even though the lower threshold employed by HWMP-TX is 0.25 , this value does not have any impact on the achieved throughput. This is because the threshold values are just to facilitate the characterization

of malicious activity.

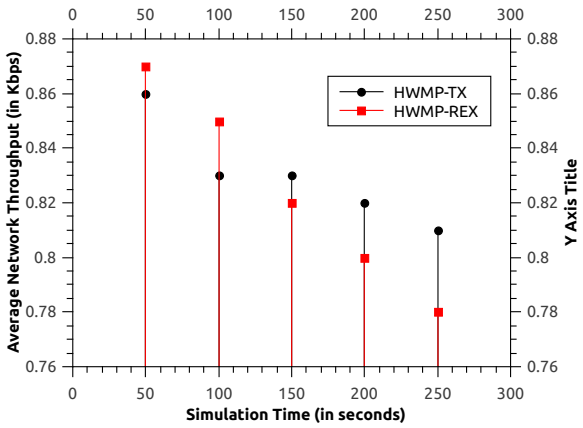


Figure 2: Throughput of HWMP-TX compared with HWMP-REX

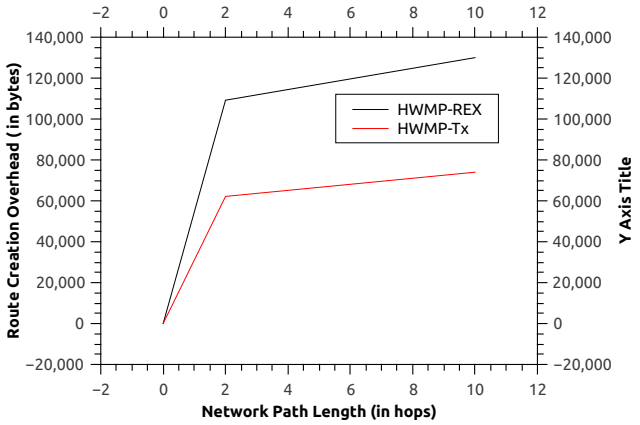


Figure 3: Overhead of HWMP-TX compared with HWMP-REX

A higher lower threshold allows protocol to converge quickly thus identifying malicious behavior. This may sometimes results into falsely ignoring genuine nodes. For a higher percentage of malicious nodes, HWMP-TX incurs higher losses, as it takes more time to converge. This behavior can be attributed to the optimistic nature of HWMP-TX protocol.

7.2 Route Creation Overhead

Next, we compute the route creation overhead of both the protocols. The route creation overhead is computed in accordance with theoretical results presented in [9]. The route creation rate per node is set to 1(λ =1) in our simulation. The average length of the route is varied between 2 to 10. The results are shown in Figure 3. The higher overhead of HWMP-REX can be attributed to increased size of PREQ. The PREQ packet employed by HWMP-REX needs to accommodate the reputation values and node addresses of each of it's neighbors. For an average of node degree 4, the packet size increases by 28 bytes (i.e.4*6 bytes per node address + 4*1 byte per reputation value). Since, HWMP-TX does not add any additional information to PREQ, its overhead remains same as HWMP. Figure 3 shows that the route creation overhead of HWMP-REX is relatively higher than HWMP-TX.

8 Discussion

Herein, we discuss the various factors that need to be accounted for comparing of both the frameworks. The comparison of both the protocols is presented in Table 4.

8.1 Path Fluctuations

Path fluctuations are frequent in HWMP-REX as it naturally prefers paths containing nodes with relatively higher reputation. The lower bounds established by HWMP-TX to tolerate transient changes in behavior thus preventing frequent switching of paths. Moreover, the paths selected by HWMP-TX perform better on average than HWMP-REX.

8.2 False Positives

False positive is a situation where a genuine is reported to be malicious. HWMP-REX directly does not exhibit such behavior, but penalizing and avoiding a good node from path selection is an indirect indication of false positive. Chances of arising such alarms are higher in HWMP-REX as it does not consider transient losses in the network. Frequent path switching is one of the indicators of false positives. On the other hand, HWMP-TX considers those packet losses and avoids frequent path fluctuations there by considerably lowering the probability of generating such false positives.

Table 4: Comparison of HWMP-REX and HWMP-TX

	HWMP-REX	HWMP-TX
Path Fluctuations	High	Low
False Positives	High	Low
False Negatives	Low	Low
Complexity	High	Low

8.3 False Negatives

False negative is a situation where a node is actually malicious and it is reported to be genuine. In HWMP-REX the watchdog module allows a node to ensure that its neighbor genuinely forwards the packets addressed to it. If the watchdog fails to detect any kind of malicious packet drop, then there is a chance of considering a malicious

node as genuine. But, as the failure of a watchdog is highly unlikely, false negatives are next to nil in HWMP-REX. In HWMP-TX, a malicious node can fake genuine behavior until it meets the established lower bounds on trust levels. Once, the trust value of a node falls below a predefined threshold, it cannot fake honest behavior. The monitoring mechanism of both the protocols ensure that a malicious node is never reported as genuine.

8.4 Complexity

HWMP-REX is more complex than HWMP-TX mainly due to the oversize of the RREQ. It also adds additional complexity to the processing of RREQ. Each node along the path to the destination appends the reputation of its neighbors along with their 6 byte addresses, thus increasing the overall size of the RREQ. Ignoring common neighbors and assuming average neighbor degree in a network to be 4, the additional overhead contributed by each node is 28 bytes.

9 Conclusion and Future Work

The performance requirements play a major role in modelling trust for a distributed network like WMN. The existing trust models for distributed networks are not directly employable and need to be significantly modified to meet the performance requirements of WMN. Therefore, we first modified an existing reputation framework, AODV-REX, specially tailored to WMN. We then proposed a complete trust model based on an alternate mechanism to employ trust in path selection process to improve the reliability and quality of selected paths. The proposed trust model allows a node to evaluate the behavior of its neighbors periodically. The experimental results of both the protocols confirm that HWMP-TX achieves better performance over existing models and also incurs less overhead. For future work, we plan to refine the established bounds to differentiate malicious behavior more accurately.

Acknowledgments

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

References

[1] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the 1997 ACM Workshop on New Security Paradigms*, pp. 48–60, 1998.

[2] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[3] M. Bahr, "Proposed routing for IEEE 802.11 s WLAN mesh networks," in *Proceedings of the 2nd Annual ACM International Workshop on Wireless Internet*, pp. 5, 2006.

[4] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR),*, Technical Report, 2003.

[5] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[6] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual ACM International Conference on Mobile Computing and Networking*, pp. 114–128, 2004.

[7] T. Eissa, S. A. Razak, R. H. Khokhar, and N. Samian, "Trust-based routing mechanism in manet: design and implementation," *Mobile Networks and Applications*, vol. 18, no. 5, pp. 666–677, 2013.

[8] T. Ghosh, N. Pissinou, and K. Makki, "Collaborative trust-based secure routing against colluding malicious nodes in multi-hop ad hoc networks," in *29th Annual IEEE International Conference on Local Computer Networks*, pp. 224–231, 2004.

[9] P. Jacquet and L. Viennot, "Overhead in mobile adhoc network protocols," 2000.

[10] I. Khalil, S. Bagchi, and N. B. Shroff, "Liteworp: A lightweight countermeasure for the wormhole attack in multihop wireless networks," in *Proceedings of IEEE International Conference on Dependable Systems and Networks (DSN'05)*, pp. 612–621, 2005.

[11] C. E. Koksal and H. Balakrishnan, "Quality-aware routing metrics for time-varying wireless mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 1984–1994, 2006.

[12] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking*, pp. 255–265, 2000.

[13] R. Matam and S. Tripathy, "Wrsr: Wormhole-resistant secure routing for wireless mesh networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1–12, 2013.

[14] K. Meka, M. Virendra, and S. Upadhyaya, "Trust based routing decisions in mobile ad-hoc networks," in *Proceedings of the Workshop on Secure Knowledge Management (SKM'06)*, 2006.

[15] Working Group of the IEEE 802 Committee et al, "IEEE p802. 11s/d5. 0–draft standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications–amendment 10: Mesh networking", 2010.

[16] F. Oliviero, L. Peluso, and S. P. Romano, "Refacing: An autonomic approach to network security based on multidimensional trustworthiness," *Computer Networks*, vol. 52, no. 14, pp. 2745–2763, 2008.

[17] F. Oliviero and S. P. Romano, "A reputation-based metric for secure routing in wireless mesh networks," in *IEEE Global Telecommunications Conference (GLOBECOM'08)*, pp. 1–5, 2008.

[18] S. Paris, C. Nita-Rotaru, F. Martignon, and A. Capone, "EFW: A cross-layer metric for reliable routing in wireless mesh networks with selfish participants," in *Proceedings of IEEE INFOCOM*, pp. 576–580, 2011.

[19] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-demand Distance Vector (AODV) Routing*, Technical Report, 2003.

[20] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. B. Royer, "A secure routing protocol for ad hoc networks," in *Proceedings of 10th IEEE International Conference on Network Protocols*, pp. 78–87, 2002.

[21] P. Subhash and S. Ramachandram, "Preventing wormholes in multi-hop wireless mesh networks," in *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT'13)*, pp. 293–300, 2013.

[22] A. P. Subramanian, M. M. Buddhikot, and S. Miller, "Interference aware routing in multi-radio wireless mesh networks," in *2nd IEEE Workshop on Wireless Mesh Networks (WiMesh'06)*, pp. 55–63, 2006.

[23] S. Tan, X. Li, and Q. Dong, "Trust based routing mechanism for securing oslr-based manet," *Ad Hoc Networks*, vol. 30, pp. 84–98, 2015.

[24] Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solution in ad hoc networks," in *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, vol. 14, 2003.

[25] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *IEEE Workshop on Wireless Mesh Networks (WiMesh'05)*, 2005.

[26] P. W. Yau, C. J. Mitchell, et al., "2HARP: A secure routing protocol for mobile ad hoc networks," 2015. (`https://pure.royalholloway.ac.uk/portal/files/4624400/2asrpf.pdf`)

P. Subhash received his M.Tech degree in Software Engineering from Jawaharlal Nehru Technological University Hyderabad, India in 2008. He is currently working towards his Ph.D degree in Wireless Mesh Network Security at Jawaharlal Nehru Technological University Hyderabad, India. His current research interest includes wireless network security and Peer-to-Peer Networking.

S. Ramachandram received the M.Tech degree from Osmania University, Hyderabad, India, in 1985, and Ph.D. degree in Computer Science and Engineering, Osmania University, Hyderabad, India in 2005. Currently he is a professor at Osmania University, Hyderabad, India. His research interests include Mobile Computing, Network Security and Grid Computing.

A Strongly Secure Certificateless Digital Signature Scheme in The Random Oracle Model

Mohammed Hassouna[1], Eihab Bashier[2,3], and Bazara Barry[3]
(Corresponding author: Eihab Bashier)
Computer Studies, National Ribat University[1]
P.O. Box 55, Khartoum, Sudan
Department of Mathematics, Physics and Statistics, College of Arts and Sciences, Qatar University[2]
P.O. Box 2713, Doha, Qatar
Department of Computer Science, Mathematical Sciences, University of Khartoum[3]
P.O. Box, 321, Khartoum, Sudan
(Email: ebashier@qu.edu.qa)
(Received Sep. 20, 2015; revised and accepted Nov. 16 & Dec. 7, 2015)

Abstract

The main purpose of this paper is to provide a security proof for the certificateless digital signature scheme found in [Hassouna, Bashier, and Barry, A short certificateless digital signature scheme, *International Conference of Digital Information Processing*, Data Mining and Wireless Communications, 2015, pp. 120–127] in the random oracle model. Two types of attacks are considered: The first type can be carried out by an outsider attacker and referred to as Type I, whereas the second one can be carried out by a malicious KGC and referred to as Type II. The possible oracles for each of the two types of attacks are discussed, and hence, the security of the proposed digital signature scheme was proved in the random oracle model.

Keywords: Certificateless cryptography, certificateless signature, pairings in elliptic curves, public-key replacement attack

1 Introduction

In 2003, Al-Riyami and Paterson [1] introduced the concept of Certificateless Public Key Cryptography (CL-PKC) to overcome the key escrow limitation of the identity-based public key cryptography (ID-PKC). In CL-PKC a trusted third party called Key Generation Center (KGC) supplies a user with a partial private key. Then, the user combines the partial private key with a secret value (that is unknown to the KGC) to obtain his/her full private key. In this way, the KGC does not know the user's private key. Then the user combines his/her secret value with the KGC's public parameters to compute his/her public key.

Al-Riyami and Paterson [1] proved that their certificateless encryption scheme is secure against fully-adaptive chosen ciphertext attack (IND-CCA). They also proposed a certificateless digital signature scheme along with certificateless key agreement protocol and hierarchal certificateless encryption scheme (HCL-PKE). Even after using the binding technique, the scheme does not have trust level 3 according to Girault's [11] definition.

Since Al-Riyami and Paterson original CL-PKC scheme was proposed [1], many certificateless cryptography schemes have appeared in literature. These schemes include the uses of certificateless encryption [7, 14], certificateless signatures [16, 19, 20] and certificateless signcryption [15, 17, 18].

Hassouna et al. [12] introduced an integrated certificateless public key infrastructure model. That model used a different key generation technique with a different binding method from Al-Riyami and Paterson [1] model. The integrated certificateless public key infrastructure model provided many practical features, like two-factor private key authentication, private key recovery, private key portability and private key archiving. These features were provided because Hassouna et al. [12] separated the process of generating private key from the process of generating the public key.

The binding technique that was proposed by Hassouna et al. [12] provided a more robust way to link the user's identity with his/her public/private keys. Furthermore, the binding technique raised very important and non-mentioned feature: it made the CL-PKC resistant to the public key replacement attack that can be done by the KGC or any adversary in case of sending the user's partial private key in an insecure channel. This was because the user's full private key is generated from a different secret value that used in the user's public key calculation.

In 2015, Hassouna et al. [13] extended their origi-

nal model that was proposed in [12], by proposing a new strong and efficient certificateless digital signature scheme. They verified its consistency and efficiency.

Furthermore, Hassouna et al. [13], proposed a new different security model that was suitable for their proposed signature scheme. In their proposed security model, the definitions of Type I and Type II adversaries had become different from the definitions introduced by Xiong et al. in [19]. However, Hassouna et al. [13] stated that their signature scheme was provably secure against their proposed security model in the Random Oracle Model (ROM), but no security proof was provided.

The main purpose of this paper is to prove the security of Hassouna et al. [13] certificateless digital signature scheme against their proposed security model. The security scheme that was introduced in [13] was based on two mathematical hard problems, namely the Computational Diffie-Hellman Problem (CDHP) and the Bilinear Diffie-Hellman Problem (BDHP) in addition to using a set of predefined hash functions. Therefore, we will prove its security in the Random Oracle Model (ROM).

The rest of this paper is organized as follows. Section 2 gives backgrounds about pairing in elliptic curves and its related cryptographic primitives. Hassouna et al. [13] digital signature scheme and their security model are in Section 3. In Section 4, we state the security proof of Hassouna et al.'s [13] signature scheme. Finally, Section 5 concludes the paper.

2 Backgrounds

In this section, we give backgrounds about pairing in eliptic curves and its related cryptography primitives that are used in this paper. Here, G_1 denotes an additive group of prime order q (particulary elliptic curve group) and G_2 a multiplicative group of the same order. We let P denote a generator of G_1.

Definition 1. Elliptic Curve Computational Diffie-Hellman Problem (ECDHP)*: Given (P, aP, bP) in G_1 where $a, b \in \mathbb{Z}_q^*$, compute abP.*

2.1 Pairing in Elliptic Curve

A pairing is a map $e : G_1 \times G_1 \longrightarrow G_2$ with the following properties:

1) The map e is bilinear: given $Q, W, Z \in G_1$, we have: $e(Q, W + Z) = e(Q, W) \cdot e(Q, Z)$ and $e(Q + W, Z) = e(Q, Z) \cdot e(W, Z)$.
   Consequently, for any $a, b \in Z_q$, we have $e(aQ, bW) = e(Q, W)^{ab} = e(abQ, W)$, etc.
2) The map e is non-degenerate: $e(P, P) \neq 1_{G_2}$.
3) The map e is efficiently computable.

Definition 2. BDH Parameter Generator: *As in [4], a randomized algorithm $\mathcal{G}$ is a BDH parameter generator if $\mathcal{G}$:*

1) *takes security parameter $k \geq 1$,*
2) *runs in polynomial time in k, and*
3) *outputs the description of groups G_1, G_2 of prime order q and a pairing $e : G_1 \times G_1 \longrightarrow G_2$.*

Formally, the output of the algorithm $\mathcal{G}(1^k)$ is (G_1, G_2, e). Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field.

We refer to [2, 3, 4, 5, 6, 8, 9, 10] for a more comprehensive description of how these groups, pairings and other parameters should be selected in practice for efficiency and security.

Definition 3. Bilinear Diffie-Hellman Problem (BDHP): *Let G_1, G_2, P and e be as above. The BDHP in G_1, G_2, e is as follows: Given P, aP, bP, cP with uniformly random choices of $a, b, c \in Z_q^*$, compute $e(P, P)^{abc} \in G_2$. An algorithm A has advantage ϵ in solving the BDHP in G_1, G_2, e if:* $\mathbf{Pr}[A(P, aP, bP, cP) = e(P, P)^{abc}] = \epsilon$.

Here, the probability is measured over the random choices of $a, b, c \in Z_q^*$ and the random bits of A.

3 Hassouna et al's Certificateless Digital Signature Scheme

In this section, we state the certificateless digital signature scheme that was proposed by Hassouna et al. [13].

- **Setup (running by the KGC):** The KGC chooses a secret parameter k to generate G_1, G_2, P, e where G_1 and G_2 are two groups of a prime order q, P is a generator of G_1 and $e : G_1 \times G_1 \longrightarrow G_2$ is a bilinear map. The KGC randomly generates the system's master key $s \in \mathbb{Z}_q^*$ and computes the system public key $P_{pub} = sP$. Then the KGC chooses cryptographic hash functions H_1 and H_2, where $H_1 : \{0,1\}^* \longrightarrow G_1$ (Map-to-Point hash function), and $H_2 : \{0,1\}^n \longrightarrow \mathbb{Z}_q^*$ (any cryptographic hash function like MD5 or SHA family). Finally, the KGC publishes the system parameters $params = < G_1, G_2, e, P, P_{pub}, H_1, H_2, n >$, while the secret master-key is saved and secured by the KGC.
- **Set-Secret-Value (running by the user):** The user m with the identity ID_m downloads the system parameters, picks two random secret values $x_m, x'_m \in \mathbb{Z}_q^*$. Then, user m computes $X_m = x'_m P$ and sends X_m to the KGC. The proposed scheme enforces the user to choose a strong password $pass$, the system at client hashes the password to be $z_m = H_2(pass)$, multiplies the base point P by the hashed password to be $z_m P$, uses the hashed value z_m as key encrypt the secret value x_m and generates the Password-based Encryption Code(PEC) as $PEC_{z_m}(x_m)$, sends copy of it to the KGC's public directory and stores copy of it along with the point $z_m P$ locally.

- **Partial-Private-Key-Extract (running by the KGC):** On receiving X_m computed by user m with identity ID_m, the KGC first computes $Q_m = H_1(ID_m)$, then it generates the partial private key of user m as $D_m = sQ_m$.

- **Set-Public-Key (running by the user):** The user m with identity ID_m computes $Q_m = H_1(ID_m)$, $Y_m = x'_m Q_m$ and sets $< X_m, Y_m >$ as his/her long-term public key P_m. Finally, user m sends Y_m to the KGC.

- **Set-Private-Key:** User m's private key is $S_m = (x_m + z_m)D_m = (x_m + z_m)sQ_m = (x_m + z_m)sH_1(ID_m)$. Also, the user generates the secret term $Z_m = x_m P$.

- **Sign:** The user generates the signature of the message M using his/her secret terms $\{x_m, Z_m\}$ as follows:

  1) The signer generates big random integer $a \in G_2^*$.
  2) The signer calculates $MP_m = H_1(m) \in G_1^*$.
  3) The signer calculates $MP_{1m} = ax_m MP_m \in G_1^*$.
  4) The signer calculates $s_m = e(MP_m, Z_m)^{ax'_m} = e(MP_m, P)^{ax_m x'_m}$.
  5) The signer sends $\sigma = (m, MP_{1m}, s_m)$ as the signature.

- **Verify:** After receiving the signature $\sigma = (m, MP_{1m}, s_m)$, the verifier uses user m's public key $< X_m, Y_m >$ to verify the signature as follows:

  1) The verifier checks whether $e(X_m, Q_m) = e(Y_m, P)$. If it holds then user m's public key is authentic, otherwise the signature is rejected.
  2) The verifier calculates $MP'_m = H_1(m) \in G_1^*$.
  3) If $MP_{1m} = MP'_m$ or $s_m = e(H_1(m), X_m)$ then the verifier rejects the signature.
  4) Otherwise, the verifier calculates $r_m = e(MP_{1m}, X_m)$.
  5) The verifier accepts the signature iff $r_m = s_m$, otherwise he/she rejects the signature.

3.1 Hassouna et al.'s Security Model

In Hassouna et al. [13] two types of adversaries were considered: Type I and Type II adversaries according to the term Z_m as follows:

1) **Type I Adversary**

   A_I which is allowed to replace the term Z_m by a valid value of his/her choice, but is not allowed to replace users' public keys and has not access to the master secret key s.

2) **Type II Adversary**

   A_{II} which has access to the master secret key s, is allowed to replace users public keys with valid values of his/her choice, but is not allowed to replace the term Z_m.

Type I adversary represents outsider attacker and Type II attacker is a malicious KGC. Two games are defined as follows.

- **Game I.** The first game is performed between a challenger C and a Type I adversary A_I as follows.

  1) Setup. The challenger C runs Setup algorithm and generates a master secret key msk and public system parameters params. C gives params to A_I, while keeping msk secret.
  2) Queries. A_I may adaptively issue the following queries to C.
     - Partial private key queries: Upon receiving a partial private key query for an identity ID, C returns the partial private key with respect to identity ID to A_I.
     - Public key queries: Given an identity ID, C returns the corresponding public key terms $< X_A, Y_A >$ to A_I.
     - Replace public key: Given an identity ID with a pair of values (x'^1_{ID}, pk^1_{ID}) which are chosen by A_I, C updates the user ID original secret/public key (x'_{ID}, pk_{ID}) to the new (x'^1_{ID}, pk^1_{ID}).
     - $Z-key$ Extraction queries: This is a new oracle in this security model, given an identity ID, C returns the corresponding $Z-key$ value Z_{ID}.
     - Replace $Z-key$: This is a new oracle in this security model which on input (ID, x^1_{ID}, Z^1_{ID}), C replaces the user ID original term (x_{ID}, Z_{ID}) by (x^1_{ID}, Z^1_{ID}).
     - Private key queries. Upon receiving a private key query for an identity ID, C returns the corresponding private key sk_{ID} to A_I.
     - Sign queries: Proceeding adaptively, A_I can request signatures on any messages m with respect to an identity ID. C computes signature, and returns to A_I.
  3) Forgery. Eventually, A_I outputs a certificateless signature σ^* on message m^* corresponding to public key pk_{ID^*} for an identity ID^*. A_I wins the game if Verify($params$, ID^*, pk_{ID^*}, m^*, σ^*) = 1 and the following conditions hold:
     - A_I has never been queried Partial private key oracle on ID^*.
     - A_I never replaced the user ID^*'s public key.
     - A_I has never been queried Private key oracle on ID^*.

– A_I has never been queried Sign oracle on (ID^*, m^*).

The success probability of A_I is defined as the probability that it wins in Game I.

- **Game II.** This game is performed between a challenger C and a Type II adversary A_{II} as follows.

  1) Setup. The challenger C runs A_{II} on k and a special Setup, and returns a master secret key msk and public system parameters params to A_{II}.
  2) Queries. In this phase, A_{II} can adaptively access the Private key oracle, Public key oracle, Replace public key oracle, $Z-key$ oracle, Replace $Z-key$ oracle and Sign oracle, which are the same as that in Game I.
  3) Forgery. A_{II} outputs a certificateless signature σ^* on message m^* corresponding to public key pk_{ID^*} for an identity ID^*. A_{II} wins the game if Verify$(params, ID^*, pk_{ID^*}, m^*, \sigma^*) = 1$ and the following conditions hold:
     – A_{II} has never been queried Private key oracle on ID^*.
     – A_{II} has never been queried Replace $Z-key$ oracle on ID^*.
     – A_{II} has never been queried Signature oracle on (ID^*, m^*).

  The success probability of A_{II} is defined as the probability that it wins in Game II.

Accordingly, the security definitions of any certificateless digital signature scheme in the Random Oracle Model (ROM) can be given as follows.

Definition 4. *A certificateless signature scheme is* $(t, q_H, q_e, q_z, q_{sk}, q_{pk}, q_s, \epsilon)$*-existentially unforgeable against Type I adversary under adaptively chosen message attacks if no t-time adversary* A_I*, making at most* q_H *to the random oracles,* q_e *partial private key queries,* q_z *to the* $Z-key$ *queries,* q_{sk} *private key queries,* q_{pk} *public key queries and* q_s *signature queries, have a success probability at least* ϵ *in Game I.*

Definition 5. *A certificateless signature scheme is* $(t, q_H, q_z, q_{sk}, q_{pk}, q_s, \epsilon)$*-existentially unforgeable against Type II adversary under adaptively chosen message attacks if no t-time adversary* A_{II}*, making at most* q_H *to the random oracles,* q_z *to the* $Z-key$ *queries,* q_{sk} *private key queries,* q_{pk} *public key queries and* q_s *signature queries, have a success probability at least* ϵ *in Game II.*

Definition 6. *A certificateless signature scheme is existentially unforgeable under adaptively chosen message attack (EUF-CMA), if the success probability of any polynomially bounded adversary in the above two games is negligible.*

4 Security Analysis

The main interesting security feature in the Hassouna et al.'s [13] signature scheme, is that its security does not depends on the security of the KGC, because the master secret of the KGC is not involved directly in the signature generation/verification. This way, the such certificateless signature schemes can enjoy the same security feature as the traditional signature scheme that are based on PKI.

This is because in the PKI context, the private key of the CA does not impact the security of the signatures that are generated by the users, and that is because the users' private keys are not connected directly with the public/private key of the CA, and the public/private key of the CA is just used to ensure the authenticity of the users by signing the users' certificates.

Furthermore, the security of Hassouna et al.'s [13] signature scheme depends on the the term Z_m which is considered as one of the private keys of the user m. The term Z_m links the user's public/private keys and any compromise in the user's public key leads to compromise in term Z_m and hence in the signature scheme.

Thinking this way, the certificateless schemes can have better chances in securing real applications, because this approach will reduce the risk of trusting the KGC without decreasing the features of the certificateless cryptography as concept, i.e eliminating the certificates and some of its management problems and also eliminating the risk of trust on the KGC.

Now we state the general definition of the security of Hassouna et al.'s [13] signature scheme in the random oracle model (ROM) given that the Adversary A has access to the oracles that have been described later.

Theorem 1. Hassouna et al.'s [13] short CLS scheme is secure against existential forgery under adaptively chosen message attacks in the random oracle model with the assumptions that the ECDHP (Elliptic Curve Computational Diffie-Hellman Problem) and BDHP (Bilinear Diffie-Hellman Problem) in G_1 are intractable.

The proof of Theorem 1 is based on the following two lemmas.

Lemma 1. Let A_I be a Type I Adversary in Game I that (t, ϵ)-breaks the proposed CLS scheme. Assume that A_I makes q_H queries to a random oracle H_1, q_e queries to the partial-private-key extraction oracle, q_z queries to the $Z-key$ extraction oracle, q_{sk} queries to the private-key extraction oracle, q_{pk} queries to the public-key request oracle and q_s queries to signing oracle and can replace $Z-key$ of any user. A_I cannot replace the public key of the challenged user and does not have the master secret. Then, there exists a (ϵ', t')-algorithm C that is able to solve the BDHP problem in group G_1, G_2 where $\epsilon' < \epsilon(\frac{q_H-1}{q_H})^{q_e+q_{sk}+q_s}$, $t' < t + (q_s+q_z)t_{sm} + q_s t_p$, t_{sm} denotes the cost of the scalar multiplication in G_1 and t_p the cost of calculating one bilinear pairing operation.

Lemma 2. Let A_{II} be a Type II Adversary in Game II that (t, ϵ)-breaks the proposed CLS scheme. Assume that A_{II} makes q_H queries to random oracles H_1, q_z queries to the $Z - key$ extraction oracle, q_{sk} queries to the private-key extraction oracle, q_{pk} queries to the public-key request oracle, q_s queries to signing oracle and can replace the public key of any user. A_{II} cannot replace $Z - key$ of the challenged user but have the master secret. Then, there exists a (ϵ', t')-algorithm C that is able to solve the ECDHP problem in group G_1 where $\epsilon' < \epsilon(\frac{q_H - 1}{q_H})^{q_{sk}+q_s}$ and $t' < t + (q_s + q_z)t_{sm} + q_s t_p$.

4.1 Proof of Lemma 1

Suppose that C is given a challenge: given $Z_m = x_m P$, abP and $X_m = r_m P$ compute $e(P, P)^{abx_m r_m}$ after interacting with A_I. Now C and A_I play the role of the challenger and the adversary respectively. C will interact with A_I as follows:

- **Setup:** C runs algorithm Setup, chooses a generator P and sets $P_{pub} = sP$, where s is the system master key, which is unknown to C. C picks an identity ID^* at random as the challenged ID in this game, and gives $params =< P, P_{pub}, H_1 >$ to A_I as the public parameters. For simplicity, we assume that for any ID_i, A_I queries H_1 before ID_i is used as an input of any query Public-key Extraction, Partial-private-key Extraction, Private-key Extraction and Signing oracles.

- **H_1-Queries:** C maintains a hash list H_1^{list} of tuple (ID_i, Q_i) as explained below. The list is initially empty. When A_I makes a hash oracle query on ID_i, if the query ID_i has already appeared on the H_1^{list}, then the previously defined value is returned. Otherwise, C chooses a random integer $a \in \mathbb{Z}_q^*$ and sets $Q_i = aP$, inserts the pair (ID_i, Q_i) in the list H_1^{list} and returns it to the adversary A_I.

- **Partial-private-key Extraction Queries:** C maintains a list E^{list} of tuple (ID_i, Q_i, D_i) which is initially empty. For any given identity ID_i, C recovers the corresponding tuple (ID_i, Q_i) from the list H_1^{list}, if $ID_i \neq ID^*$ then sets $D_i = sQ_i$ and returns it to the adversary A_I and adds (ID_i, Q_i, D_i) to the list E^{list}. Otherwise($ID_i = ID^*$), C aborts and outputs "failure"(denote this event by E_1).

- **Public-key Extraction Queries:** C maintains a list pk^{list} of tuple (ID_i, Q_i, r_i, pk_i) which is initially empty. When A_I queries on input ID_i, C checks whether pk^{list} contains a tuple for this input. If it does, the previously defined value is returned. Otherwise, C recovers the corresponding tuple (ID_i, Q_i) from the list H_1^{list} and picks a random value $r_i \in \mathbb{Z}_q^*$, computes $pk_i =< X_i, Y_i >=< r_i P, r_i Q_i >$ and returns pk_i. Then, adds (ID_i, Q_i, r_i, pk_i) to the list pk^{list}.

- **$Z - key$ Extraction Queries:** C maintains a list Z^{list} of tuple (ID_i, Z_i) which is initially empty. if Z^{list} already contains the pair (ID_i, Z_i), then it returns it to the adversary A_I, otherwise C calls the oracle Private Key Extraction on identity ID_i and gets the value Z_i, gives it to the adversary A_I and inserts it in the list Z^{list}.

- **Private-key Extraction Queries:** C maintains the list sk^{list} for query on input ID_i. If $ID_i = ID^*$, C stops and returns "failure" (denote the event by E_2). Otherwise, C picks a random number $x_i \in \mathbb{Z}_q^*$ and performs as follows:
  - If the E^{list} and the pk^{list} contain the corresponding tuple (ID_i, Q_i, D_i) and the tuple (ID_i, Q_i, r_i, pk_i) respectively, C sets $sk_i = x_i D_i$, $Z_i = x_i P$, returns (ID_i, x_i, sk_i, Z_i) to A_I and adds them to the list sk^{list}.
  - Otherwise, C makes a partial-private-key Extraction query and a Public-key Extraction query on ID_i, then simulates as the above process, sends (ID_i, x_i, sk_i, Z_i) to A_I and adds them to the list sk^{list}.

- **$Z - key$ Replacement (ID_i, x_i', Z_i'):** When A_I queries on input (ID_i, x_i', Z_i'), C checks whether the tuple (ID_i, Z_i) is contained in the Z^{list} . If it is, C sets $Z_i = Z_i'$ and adds (ID_i, Z_i') to the Z^{list}. Here, we assume that C can obtain a replacing secret value x_i' corresponding to the replaced $Z - key = Z_i'$ from A_I. Otherwise, C executes Private Key extraction to generate (ID_i, sk_i, Z_i), then sets $Z_i = x_i' P$ and inserts it in the list Z^{list}.

- **Signing Queries:** When a signing query (ID_i, m_j) is coming, C acts as follows:
  - If $ID_i = ID^*$, C stops and returns "failure status" (denote the event by E_3).
  - Otherwise, C recovers the tuple (ID_i, x_i, sk_i, Z_i) from the sk^{list} and the tuple (ID_i, Q_i, pk_i) from the pk^{list} and the tuple (m_j, MP) from H_1^{list}.
  - Picks a random integer $a \in \mathbb{Z}_q^*$.
  - Computes $MP_1 = ax_i MP$.
  - Computes $s_i = e(MP, Z_i)^{ar_i}$ and (MP_1, s_i) is the signature for the identity ID_i on the message m_j. C returns (MP_1, s_i) to A_I as response to the signing oracle.

Finally, A_I stops and outputs a signature $\sigma = (V^*, S^*)$ on the message m^* for the identity ID^*, which satisfies the equation $\text{Verify}(m^*, ID^*, pk^*, S^*) = 1$. C recovers the tuple (ID^*, Q^*, pk) from pk^{list}, the tuple (ID^*, x^*, Z^*), (m^*, MP^*) from Z^{list} and H_1^{list} picks a random integer $a^* \in \mathbb{Z}_q^*$. Then, we have $e(V^*, X_i) = e(a^* x^* b^* P, rP) = S^*$, that is: $e(P, P)^{a^* x^* r b^*} = S^*$.

Hence C can successfully compute and output $e(P,P)^{a^*r} = S^{*1/(x^*b^*)}$ as solution to the A_I's challenge. So, C breaks the BDHP problem in G_1, G_2. Now we analyze the advantage of C in this game.

Note that the responses to A_I's H_1 queries are indistinguishable from the real life. Since each response is uniformly random and independently distributed in G_1^*. The responses of queries H_1 provided for A_I are all valid. The responses of Partial-private-key extraction queries, Private-key extraction queries and signing queries are valid if the events E_1, E_2 and E_3 never happen. Furthermore, if A_I forges a valid signature and events E_1, E_2 and E_3 do not happen, then C can solve the BDHP problem. Therefore, if none of the events E_1, E_2 and E_3 happens, C can solve the BDHP problem successfully. Now, Let's bound the probability for these events. From the description above we have: $Pr(\neg E_1 \wedge \neg E_2 \wedge \neg E_3) = (\frac{q_H-1}{q_H})^{q_e+q_{sk}+q_s}$.

In conclusion, challenger's C advantage is $\epsilon' < \epsilon(\frac{q_H-1}{q_H})^{q_e+q_{sk}+q_s}$ with the running time cost as $t' < t + (q_s+q_z)t_{sm} + q_s t_p$, where t_{sm} denotes the cost of the scalar multiplication in G_1 and t_p the cost of calculating one bilinear pairing operation.

4.2 Proof of Lemma 2

Suppose that C is given a challenge: given $Z_m = x_m P$ and abP, compute $abx_m P$ after interacting with A_{II}. Now C and A_{II} play the role of the challenger and the adversary respectively. C will interact with A_{II} as follows:

- **Setup:** C runs algorithm Setup, chooses generator P and sets $P_{pub} = sP$, where s is the system master key. C picks an identity ID^* at random as the challenged ID in this game, and gives $params =< P, P_{pub}, H_1 >$ and the master secret s to A_{II} as the public parameters. For simplicity, we assume that for any ID_i, A_{II} queries H_1 before ID_i is used as an input of any query Public-key Extraction, Private-key Extraction and Signing oracles.

- **H_1-Queries:** C maintains a hash list H_1^{list} of tuple (ID_i, Q_i) as explained below. The list is initially empty. When A_{II} makes a hash oracle query on ID_i, if the query ID_i has already appeared on the H_1^{list}, then the previously defined value is returned. Otherwise, C chooses a random integer $a \in \mathbb{Z}_q^*$ and sets $Q_i = aP$. Then, he inserts the pair (ID_i, Q_i) in the list H_1^{list} and returns it to the adversary A_{II}.

- **Public-key Extraction Queries:** C maintains a list pk^{list} of tuple (ID_i, Q_i, r_i, pk_i), which is initially empty. When A_{II} queries on input ID_i, C checks whether pk^{list} contains a tuple for this input. If it does, the previously defined value is returned. Otherwise, C recovers the corresponding tuple (ID_i, Q_i) from the list H_1^{list} and picks a random value $r_i \in \mathbb{Z}_q^*$, computes $pk_i =< X_i, Y_i >=< r_i P, r_i Q_i >$ and returns pk_i. Then, C adds (ID_i, Q_i, r_i, pk_i) to the list pk^{list}.

- **Public-key Replacement** (ID_i, r_i', pk_i')**:** When A_{II} queries on input (ID_i, pk_i), C checks whether the tuple (ID_i, Q_i, r_i, pk_i) is contained in the pk^{list} . If it does, C sets $pk_i = pk_i'$ and adds (ID_i, Q_i, r_i', pk_i') to the pk^{list}. Here, we assume that C can obtain a replacing secret value r_i' corresponding to the replaced $pk_i' =< r_i' P, r_i' Q_i >$ from A_{II}. Otherwise, C executes Public Key extraction to generate (ID_i, Q_i, r_i, pk_i), then sets $pk_i = pk_i'$ and inserts it in the list pk^{list}.

- $Z-key$ **Extraction Queries:** C maintains a list Z^{list} of tuples (ID_i, Z_i), which is initially empty. If Z^{list} already contains the pair (ID_i, Z_i), then C returns it to the adversary A_{II}, otherwise C calls the oracle Private Key Extraction on identity ID_i and gets the value Z_i, forwards it to the adversary A_{II} and inserts it in the list Z^{list}.

- **Private-key Extraction Queries:** C maintains the list sk^{list}, for query on input ID_i, If $ID_i = ID^*$, C stops and outputs "failure" (denote the event by E_1). Otherwise, C picks a random number $x_i \in \mathbb{Z}_q^*$ and performs as follows:
  - If the E^{list} and the pk^{list} contain the corresponding tuple (ID_i, Q_i, D_i) and the tuple (ID_i, Q_i, r_i, pk_i) respectively, then C sets $sk_i = x_i D_i$, $Z_i = x_i P$, returns (ID_i, x_i, sk_i, Z_i) to A_{II} and adds them to the list sk^{list}.
  - Otherwise, C makes a Partial-private-key Extraction query and a Public-key Extraction query on ID_i, then simulates as the above process, sends (ID_i, x_i, sk_i, Z_i) to A_{II} and adds them to the list sk^{list}.

- **Signing Queries:** When C receives a signing query (ID_i, m_j), it acts as follows:
  - If $ID_i = ID^*$, C stops and returns "failure status" (denote the event by E_2).
  - Otherwise, C recovers the tuple (ID_i, x_i, sk_i, Z_i) from the sk^{list}, the tuple (ID_i, Q_i, pk_i) from the pk^{list} and the tuple (m_j, MP) from H_1^{list}.
  - Picks random integer $a \in \mathbb{Z}_q^*$.
  - Computes $MP_1 = ax_i MP$.
  - Computes $s_i = e(MP, Z_i)^{ar_i}$ and (MP_1, s_i) is the signature for the identity ID_i on the message m_j. C returns (MP_1, s_i) to A_{II} as response to the signing oracle.

Finally, A_{II} stops and outputs a signature $\sigma = (V^*, S^*)$ on the message m^* for the identity ID^*, which satisfies the equation $\text{Verify}(m^*, ID^*, pk^*, S^*) = 1$. C recovers the tuple (ID^*, Q^*, pk^*) from pk^{list}, the tuple (ID^*, Z), (m^*, MP^*) from Z^{list}, H_1^{list} and picks a random integer $a^* \in \mathbb{Z}_q^*$. Then, we have $e(V^*, X_i^*) = e(a^* x b^* P, r^* P) = S^*$, then $a^* b^* x P = V^*$.

Hence C can successfully compute and output $a^*b^*xP = V^*$ as solution to the A_{II}'s challenge. So, C breaks the ECDHP problem in G_1.

Also, C can solve the ECDHP problem successfully, if none of the events E_1 and E_2 happens. Now, we have:

$$Pr(\neg E_1 \wedge \neg E_2) = \left(\frac{q_H - 1}{q_H}\right)^{q_{sk}+q_s}.$$

Again, the challenger's C advantage is $\epsilon' < \epsilon(\frac{q_H-1}{q_H})^{q_{sk}+q_s}$ with a running time cost as $t' < t + (q_s + q_z)t_{sm} + q_s t_p$.

Therefore, if the attacker has no advantage in winning Game I and Game II which are defined as in Lemma 1 and Lemma 2, then the proposed certificateless digital signature scheme is existential unforgeable against adaptively chosen message attacks in the random oracle model with the assumptions that ECDHP and BDHP in G_1 are intractable.

5 Conclusions and Remarks

In this paper, the security proof of the digital signature scheme proposed by Hassouna et al. [13] was introduced in the random oracle model. The proposed signature scheme is strong, efficient, and resistant to the key-replacement attack.

Furthermore, since this proven signature scheme does not depend on the KGC master secret, then any cryptographic system utilizes this signature scheme can provide authentication and non-repudiation services even if the KGC is compromised as in the traditional PKI-based systems.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology (Asiacrypt'03)*, pp. 452–473, Springer, 2003.

[2] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *In Advances in Cryptology (Crypto'02)*, LNCS 2442, pp. 354–368, Springer, 2002.

[3] P. S. L. M. Barreto, B. Lynn, and M. Scott, "Constructing elliptic curves with prescribed embedding degrees," in *Security in Communication Networks (SCN'2002)*, LNCS 2576, pp. 263–273, Springer, 2002.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (Crypto'01)*, LNCS 2139, pp. 213–229, Springer, 2001.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[6] D. Boneh, H. Shacham, and B. Lynn, "Short signatures from the weil pairing," in *Advances in Cryptology (Asiacrypt'01)*, LNCS 2248, pp. 514–532, Springer, 2001.

[7] A. W. Dent, B. Libert, and K. G. Paterson, "Certificateless encryption schemes strongly secure in the standard model," in *Public Key Cryptography*, pp. 344–359, 2008.

[8] R. Dupont, A. Enge, and F. Morain, "Building curves with arbitrary small mov degree over finite prime fields," *Journal of Cryptology*, vol. 18, no. 2, pp. 78–89, 2002.

[9] S. D. Galbraith, "Supersingular curves in cryptography," in *Advances in Cryptology (Asiacrypt'01)*, LNCS 2248, pp. 495–513, Springer, 2001.

[10] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the tate pairing," in *5th International Symposium on Algorithmic Number Theory*, LNCS 2369, pp. 324–337, Springer, 2002.

[11] Girault, "Self-certified public keys," in *Advances in Cryptology (Eurocrypt'91)*, LNCS 547, pp. 490–497, Springer, 1992.

[12] M. Hassouna, B. Barry, N. Mohamed, and E. Bashier, "An integrated public key infrastructure model based on certificateless cryptography," *International Journal of Computer Science and Information Security*, vol. 11, pp. 1–10, 2013.

[13] M. Hassouna, E. Bashier, and B. Barry, "A short certificateless digital signature scheme," in *International Conference of Digital Information Processing, Data Mining and Wireless Communications*, pp. 120–127, 2015.

[14] S. Sharmila Deva Selvi, S. Sree Vivek, and C. Pandu Rangan, "CCA2 secure certificateless encryption schemes based on RSA," *IACR Cryptology ePrint Archive*, vol. 2010, pp. 459, 2010.

[15] S. Sharmila Deva Selvi, S. Sree Vivek, and C. Pandu Rangan, "Certificateless kem and hybrid signcryption schemes revisited," in *International Conference of Information Security, Practice and Experience (ISPEC'10)*, pp. 294–307, 2010.

[16] C. Wang, D Long, and Y. Tang, "An efficient certificateless signature from pairing," *International Journal of Network Security*, vol. 8, no. 1, pp. 96–100, 2009.

[17] W. Xie and Z. Zhang, "Certificateless signcryption without pairing," *IACR Cryptology ePrint Archive*, vol. 2010, pp. 187, 2010.

[18] W. Xie and Z. Zhang, "Efficient and provably secure certificateless signcryption from bilinear maps," in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS'10)*, pp. 558–562, 2010.

[19] H. Xiong, Z. Qin, and F. Li, "An improved certificateless signature scheme secure in the standard model," *Fundamenta Informaticae*, vol. 88, pp. 193–206, 2008.

[20] L. Zhang and F. Zhang, "A new provably secure certificateless signature scheme," in *IEEE International Conference on Communications*, pp. 1685–1689, 2008.

Mohammed Alfateh Hassouna is Assistant Professor at Department of Computer Science - Faculty of Computer Studies - The National Ribat University - Sudan. He gained his PhD in cryptography from the University of Khartoum - Sudan. Currently he is working as ICT Manger at the National Ribat University. He has many published papers in the international journals and conferences related to the information security and cryptography.

Bazara Barry is an associate professor at the department of Computer Science - University of Khartoum and formerly the head of the same department. He was director of research at the Faculty of Mathematical Sciences. Bazara is a reviewer and TPC head/member of many international journals/conferences and a member of the IEEE. He has won several best paper and research awards at the international level.

Eihab Bashier obtained his PhD in 2009 from the University of the Western Cape in South Africa. He is an associate professor of applied mathematics at University of Khartoum, since 2013 and recently, he joined the department of Mathematics, Physics and Statistics of Qatar University. The research interests of Dr. Bashier are mainly in numerical methods for differential equations with applications to biology and in information and computer security. In 2011, Dr. Bashier won the African Union and the Third World Academy of Science (AU-TWAS) young scientists national award in basic sciences, technology and Innovation. Dr. Bashier is a reviewer for many international journals and an IEEE member.

Group Rekeying Scheme for Dynamic Peer Group Security in Collaborative Networks

Depeng Li[1] and Srinivas Sampalli[2]
(Corresponding author: Srinivas Sampalli)
Department of Information and Computer Sciences, University of Hawaii at Manoa[1]
1680 East-West Road, Honolulu, HI, USA, 96822
Faculty of Computer Science, Dalhousie University[2]
6050 University Avenue, Halifax, Nova Scotia B3H 4R2 Canada
(Email: srini@cs.dal.ca)
(Received May 12, 2010; revised and accepted Jan. 10 & Nov. 10, 2013)

Abstract

Contributory group key management schemes are popularly used for dynamic peer group communications in collaborative environments. Previous contributory group key management schemes require every group member to perform a number of expensive Diffie-Hellman operations whenever the group membership changes. This is not always affordable for devices in resource-constrained networks. In this paper, we present an efficient group key management scheme, in which most group members process one way hash functions and only a few members perform Diffie-Hellman operations. Our proposal is an extension of the Tree-based Group Diffie-Hellman (TGDH) technique. Performance analyses and experimental results show that our approach achieves a new performance minimum, while guaranteeing the same level of security as other approaches.

Keywords: Dynamic peer groups, group key management, resource limited networks

1 Introduction

There has been a growing demand in the past a few years for security in collaborative environments deployed for emergency services, as well as many applications in military, business, government and research organizations [9, 15, 47]. Examples of such collaborative applications include tele/video-conferencing, white-boards, and distributed simulations. Many of these applications involve dynamic peer groups (DPGs) in which the group size is relatively small (around several hundreds of nodes) and each group member can simultaneously be the message sender and receiver [2, 14]. Group members may join or leave the group at any time. To provide security services, a common and efficient solution is to encrypt group messages with a symmetric group key shared by all group application participants. Group key management is the set of processes which supports the establishment of group keys and the maintenance of ongoing keying relationships between parties, including replacing older keys with newer ones as necessary [24]. Efficient management of group keys generating, distributing, and group rekeying whenever the group composition changes is critical to the successful implementation of the scheme in networks in general, and resource-limited networks, in particular.

Group key management schemes should ensure that the new member and the leaving member should not obtain the current group key. In other words, two requirements must be satisfied:

Forward secrecy: Previous group members who know contiguous subsets of old group keys must not be able to discover subsequent group keys after they leave the group.

Backward secrecy: New group members who know a contiguous subset of current group keys must not be able to discover preceding group keys.

Furthermore, performance-relevant requirements such as computational cost, communication overhead, fault-tolerance, and storage consumption must be considered, especially in resource-constrained networks.

A number of group key management schemes have been proposed. They can be classified into two broad categories, namely, *centralized* [31, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 46] and *contributory* [4, 5, 6, 8, 15, 16, 18, 21, 33, 36].

In a typical centralized group key management scheme, a key server is responsible for the generation, encryption, and distribution of the symmetric group key, auxiliary keys, and individual keys to all other group members. Although such a scheme has good performance, the key server can be a single point of failure/bottleneck.

In contributory group key management schemes, every group member contributes to the generation of the group key. Unlike a centralized scheme which relies on one or a few key servers, a contributory scheme is supported by all group members and therefore it is more fault tolerant than the centralized one. But most existing contributory schemes e.g. TGDH [15] display poor performance and a low level of scalability since they have to process expensive public key operations.

Recently, a number of contributory group key managements have been proposed for particular network settings such as [27] for Ad-hoc network, [22] for mobile wireless networks, and TGDH [15] for collaborative networks in DPG environments.

To provide the authentication service, some authenticated group key managements have been proposed. As one of the most popular authentication primitives, ID-based group key authentication [12] has been widely utilized to design a number of efficient authentication group key management [17, 21, 32]. They have been evaluated and analyzed by cryptanalysis [11], attacks [10, 40], and other security means regarding their security.

1.1 Motivation

Currently, deploying DPGs in wireless and mobile environments becomes an attractive choice for not only customers but also service providers. Meanwhile, advancements in wireless and mobile communication technologies together with the significant enhancement of the processing capability of communication devices (e.g. laptops and wearable computers) enable ubiquitous computing. In such networks, mobile nodes establish routes dynamically among themselves to form their own network on the fly without an existing infrastructure and thus make a good choice for DPGs.

However, previous group key management schemes [26, 47] cannot be deployed in such networks directly for several reasons. First, most mobile networks are resource-limited and lack a native infrastructure. Hence, they pose non-trivial challenges for the deployment of group key schemes. Traditional centralized schemes which rely on a key server cannot be a practical choice because of the lack of infrastructure in such networks. Second, such networks have stringent resource constraints. Some low-end mobile nodes tend to be restricted in their computational capability and cannot perform many and frequent computational-intensive operations such as public key cryptographic operations. Third, the communication bandwidth is also limited. Given these constraints, group key management schemes should be lightweight in order to conserve bandwidths, energy, storage, and computations. Our paper proposes an efficient contributory group key management scheme for dynamic peer groups.

1.2 Contributions

Our proposal TGDH+ is an extension of the Tree-based Group Diffie-Hellman (TGDH) [15]. TGDH uses a binary key tree for group key updates. We make a number of enhancements to TGDH. When group members join, our approach achieves the group keys update using a one-way hash function. When a group member leaves, it uses three efficient techniques, namely, *the auxiliary group key*, *moving the child key tree*, and *the dominating algorithm*, to reduce computational costs and communication overhead.

1.3 Assumptions and Scopes

Our proposal assumes that the reliability and message-in-order service are already provided by group communication systems, such as Extended Virtual Synchrony (EVS) [14, 25].

In this paper, we will specifically focus on developing efficient group key agreement for DPGs in collaborative network settings. Though deploying the ID-based authentication scheme [19, 29], the proposed group key management TGDH+ will not include any new authentication means which are out of the scope of this paper. Thus, we will not analyze TGDH+'s security regarding authentication in detail.

The rest of this paper is organized as follows. Notations and concepts are introduced in Section 2. Our proposal is described in Section 3. Performance analysis is given in Section 4. Experimental results are presented in Section 5. Concluding remarks are given in Section 6. Detailed performance comparison is discussed in Appendix A.

2 Preliminaries

Table 1: Notation

$\|$	Concatenation
M_i	Group member i
r_i	Random integer generated by M_i
{X}y	Plaintext X is encrypted with key y
α	Exponentiation base shared in advance
C	An integer known in advance
H(G)	To perform hash function on input G

2.1 Tree-based Group Diffie-Hellman (TGDH)

TGDH [15] is one of the most efficient contributory group key management schemes proposed in the literature. Since our proposal is an extension of TGDH, we provide an overview of the scheme here.

The crux of the group key management scheme in TGDH is to use a binary key tree for group key updates. Let **T** be a binary tree in which every node is represented by $< h, i >$ where h is its height (level) and i is its index. Each node in the binary tree, has two keys, node

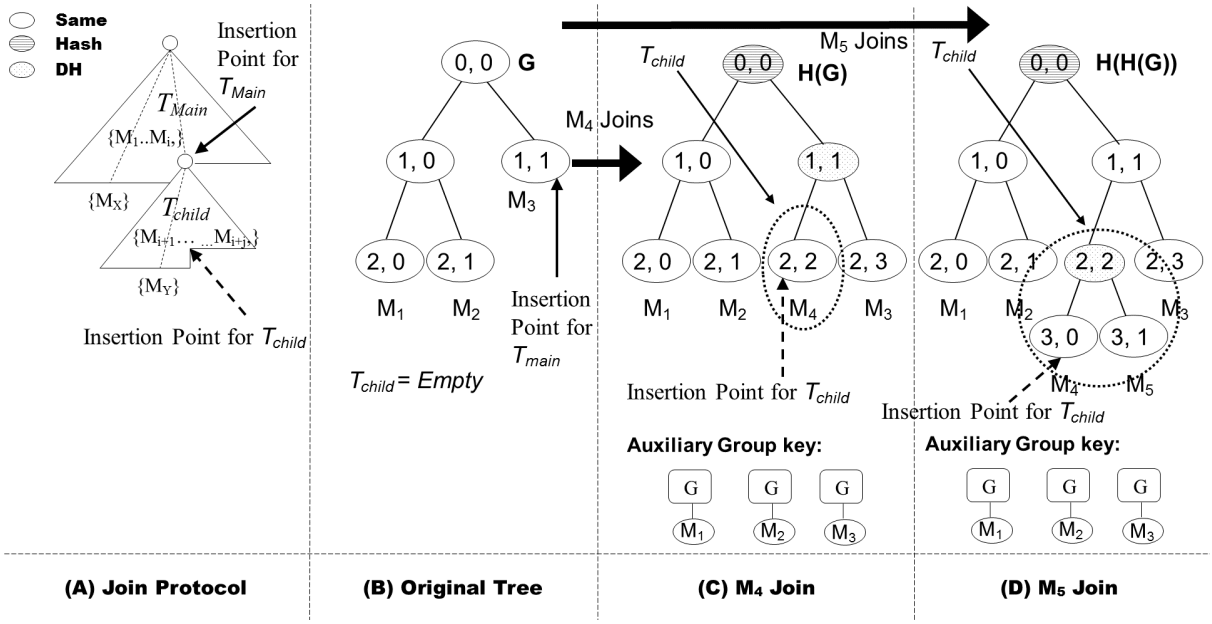


Figure 1: TGDH+: Key tree updates for group members joining – (a) Join protocol, (b) Original tree, (c) M_4 joins, and (d) M_5 joins

key (K) and blinded key (BK). The node key associated with the node $< l, v >$ is $K_{<l,v>}$ and its blinded key is $BK_{<l,v>} = \alpha^{K_{<l,v>}}$. In TGDH, every group member should be aware of the entire key tree structure.

Each node in the tree is either a leaf node or a parent node. Each leaf node represents a group member M_i. A random integer, namely, r_i, is generated specifically for M_i. This random value will be treated as the leaf node's node key. The node key of an internal/parent node $< l, v >$ is derived from the keys of its children node, $< l+1, 2v >$ and $< l+1, 2v+1 >$. This is represented by "Equation (1)" below:

$$\begin{aligned} K_{<l,v>} &= BK_{<l+1,2v>}^{K_{l+1,2v+1}} \\ &= BK_{<l+1,2v+1>}^{K_{l+1,2v}} \\ &= \alpha^{K_{l+1,2v}K_{l+1,2v+1}} \end{aligned} \quad (1)$$

The node key of the root in the tree **T** is the group key. While a new group member joins, the shallowest leftmost leaf node in the key tree is selected as the sponsor and acts as the sibling for the new group member. When a group member leaves, the sponsor is the shallowest leftmost leaf node of the sub-tree rooted as the leaving members' sibling node. The sponsor is responsible for updating its secret random integer r_i as well as all keys along the key path starting from itself and ending at the root node. Then, the sponsor multicasts all updated blinded keys, based on which, other group members could update keys on their own key paths and finally compute the new group key by themselves.

2.2 Definitions

Key path: It is a path in the key tree starting at the leaf node hosted by a group member (e.g. M_i) and ending at the key trees root. We name the key path of a group member, for instance, M_i, as KP_i. The group member (e.g. M_i) should host all node keys on the key path (e.g. KP_i) including the node key of the root which is the group key in our paper. All those node keys in the key path is called KEY_i^*.

Sibling path: For each node on a key path e.g. KP_i, there is a corresponding sibling node. All those sibling nodes construct the sibling path for a particular group member (e.g. M_i). In our paper, M_i hosts all blinded keys on its sibling path which are defined as $BKEY_i^*$.

Key sub-path: Unlike a key path, a sub-path starts at any node, N_x and ends at any other node, N_y on a key path KP_i. It is called key sub-path, namely, $KSP_{i,x,y}$. All node keys on the key sub-path $KSP_{i,x,y}$ are called $KEY_{i,x,y}^*$.

3 TGDH+ Group Key Management Scheme

In this section, we present our scheme TGDH+, an extension of TGDH. The basic idea behind our TGDH+ group key management scheme is the following. A one-way hash function **H** is used to update the group key when group members join. In contrast, the updates of Diffie-Hellman (DH)-based keys (including both node key and blinded key) resulting from the join of members have to be postponed until a group member leaves. When the leav-

ing event for a group member happens, we propose a new method which updates keys associated with the key tree. Utilizing hash functions to handle group members' joining has been suggested by some centralized group key management schemes such as ELK [31] and LKH+ [38]. However, DH-based contributory schemes have not adopted this technique since the key calculation means of "Equation (1)" cannot be align with it.

Specifically, our proposal includes three new schemes, namely, *the auxiliary group key method*, *the approach to move the child key tree*, and *the dominating algorithm*. In the auxiliary group key scheme, every group member in the main key tree stores an auxiliary group key G_a which is used as the partial key to calculate the future group key when the leaving member associates with the child key tree. The moving child key tree scheme is a method to decrease the number of updated key paths. The dominating algorithm is proposed to enable every group member to become aware of the nodes responsible for updating overlapped intermediate nodes.

In the following subsections, we describe the protocols for *join*, *leave*, *merge*, and *partition*.

3.1 Join Protocol

3.1.1 Method to Update the Key Tree Structure

The key tree shown in Figure 1 (a), includes two parts: the main key tree, T_{Main} and a child key tree, T_{child}. At the very beginning of the group key scheme, both of them are empty which means that there are no nodes available. Every key tree should have its insertion point, which is the shallowest leftmost node in the key tree.

For every group membership change, the rules below should be followed: 1) When a group member leaves or the group partitions/merges, T_{child} will merge into T_{Main} and then T_{child} is assigned as *EMPTY*. 2) When a group member joins, the method of inserting it into the key tree should be based upon whether T_{child} is *EMPTY*. If T_{child} is not *EMPTY*, the new group member should be appended to the T_{child}. Otherwise, T_{child} should be generated with its root located at the insertion point of T_{Main}. Then, T_{child} is not *EMPTY*. The remaining new join nodes should be appended into T_{child} and located at the insertion point of T_{child}. Figure 1 (a) – (d) shows a scenario in which i group members ($\mathrm{M}_1 \ldots \mathrm{M}_i$) are already within the group and, then, the following group membership events happen:

$$< M_x^{Leave}, M_{i+1}^{Join}, M_{i+2}^{Join} ... M_{i+j}^{Join}, M_y^{Leave} | where \quad j \geq 0 \rangle$$

Between the two leave requests from M_x and M_y *where* $1 \leq x \leq i$ and $1 \leq y \leq i+j$, group members M_{i+1}, $\ldots \mathrm{M}_{i+j}$ request to join one by one. Notice that this event model can represent all scenarios occurring in group membership changes due to the fact that $j \geq 0$. Thus, all event sequences can be segmented by leave events. For the remainder of this paper, this model will be utilized to demonstrate group events.

With the group membership change input, T_{child} should be *EMPTY* after M_x leaves. Then, when M_{i+1} requests to join, the join protocol generates T_{child} with the root located at the T_{Main} insertion point and the join protocol inserts M_{i+1} into T_{child}. Now T_{child} is not *EMPTY*. The current group key G is stored by every group member in T_{Main} as the auxiliary group key G_a. Subsequent join requests, $\mathrm{M}_{i+2}, \ldots \mathrm{M}_{i+j}$ can be appended into T_{child} at T_{child}'s insertion point. After M_y leaves, T_{child} is assigned to *EMPTY*.

Here are two examples. The tree shown as Figure 1 (b) is the beginning scenario. The trees shown in Figure 1 (c) and Figure 1 (d) result from the joining of M_4 and M_5, respectively. Specifically, as shown in Figure 1 (c), M_4 joins and a new leaf $< 2,2 >$ is generated to represent it. The insertion point for T_{Main} is located at node $< 1,1 >$ which should be renamed $< 2,3 >$ and works as the sponsor. Therefore, a new intermediate node $< 1,1 >$ is generated which works as both sponsor $< 2,3 >$ and the new leaf $< 2,2 >$'s parent. Every group member in T_{Main}, i.e. M_1, M_2, or M_3 should store the current group key G as its auxiliary group key: $G_a = G$.

As shown in Figure 1 (d), M_5 joins and a new leaf $< 3,1 >$ is generated to represent it. $< 3,1 >$ is appended into T_{child} rooted with $< 1,1 >$. Node $< 2,2 >$, representing member M_4, is selected as the sponsor and is renamed as $< 3,0 >$. The join protocol generates a new node $< 2,2 >$ which works as $< 3,0 >$ and $< 3,1 >$'s parents. Since when M_5 joins, T_{child} is not *EMPTY*, the auxiliary group key for every member in T_{Main} such as M_1, M_2, or M_3 stays the same.

3.1.2 Group Key Updates

For a group member join request from $\mathrm{M}_{i'}$, the proposed join protocol selects the sponsor S in the same manner as TGDH. However, the difference between TGDH and the proposed approach is that every group member updates the current group key, G with $\mathbf{H}$(G) rather than updating all keys associated with the nodes on sponsor S's key path, where $\mathbf{H}$ is a secure one-way hash function. Then, S and $\mathrm{M}_{i'}$ initiate a 2-party DH key exchange scheme to generate the shared key K, which works as the node key of S and $\mathrm{M}_{i'}$'s parent node. Finally, S sends $\mathrm{M}_{i'}$ the encrypted current group key, $\{\mathbf{H}(\mathrm{G})\}\mathrm{K}$ and $\mathrm{M}_{i'}$ decrypts the ciphertext with key K to obtain the current key, $\mathbf{H}$(G).

For example, in Figure 1 (c), M_4 joins and M_3 is selected as the sponsor. It refreshes its secret random r_3 with a new random value, r_3' and calculates the updated blinded key of its leaf node, $BK'_{<2,3>} = \alpha^{r_3'}$. Then M_3 and the new group member M_4 launch a 2-party DH to calculate a shared key, $K_{<1,1>}$. M_3 sends $C = \{BKEY_3^* || BK'_{<2,3>} || \{G'\}K_{<1,1>}\}$ to M_4 *where* $G' = \mathbf{H}(G)$. M_4 calculates $\mathrm{K}_{<1,1>}$ and then decrypts the ciphertext C to obtain the new group key, G'. Other members can calculate the new group key, G', via a secure hash function $\mathbf{H}$ since they all know the current group key, G.

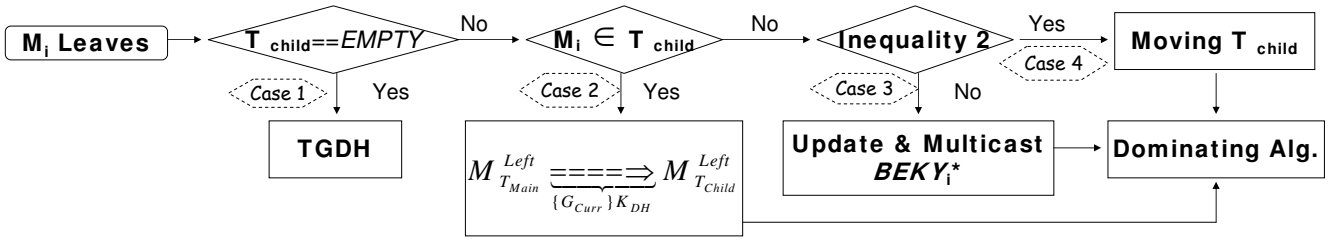


Figure 2: TGDH+: Outline of leave protocol

In Figure 1 (d), when a new group member M_5 joins, as the shallowest leftmost node in the child key tree T_{child}, M_4 is selected as the sponsor. It refreshes its secret random r_4 with a new random value r_4' and then calculates the updated blinded key of its leaf node, $BK'_{<3,0>} = \alpha^{r_4'}$. Then M_4 and the new group member M_5 launch a 2-party DH to calculate a shared key, $K_{<2,2>}$. M_4 sends $C = \{BKEY_4^* || BK'_{<3,0>} || \{G''\}K_{<2,2>}\}$ to M_5 *where* $G'' = \mathbf{H}(\mathbf{H}(G))$. M_5 first calculates $K_{<2,2>}$ and then decrypts the ciphertext C to obtain the new group key G''. Other current group members could also calculated the new group member.

Notice that the mutual authentication between the sponsor and the new group member will deploy technologies such as certifications [24] or the ID-based authentication [21] which are already mature.

3.2 Leave Protocol

3.2.1 Strategy for Updating Key Tree Structure

Suppose that group member M_i, who is represented by the leaf $< h, i >$, leaves the group. Figure 2 shows the outline of the leave protocol for TGDH+.

If T_{child} is *EMPTY*, call it *Case 1*. The proposed leave protocol is as same as that for TGDH.

If T_{child} is NOT *EMPTY* and $< h, i >$ is within T_{child}, call it *Case 2*. The key tree structure stays the same.

If T_{child} is NOT *EMPTY* and $< h, i >$ is not within T_{child}, there are two cases: either moving T_{child} or not moving. The former is shown in Figure 3 (a).

Whether T_{child} should be moved or not depends on both the leaf node $< h, i >$'s position and computational cost. Inequality (2) decides which one is more efficient, moving T_{child} or not. The left side of Inequality (2) demonstrates the computation cost for moving the T_{child}: it includes the cost to update keys associated with all nodes both in T_{child} and in key sub-path $\mathrm{KSP}_{i,x,r}$ (starting at node x, the root of T_{child} and ending at the node r, root of the key tree). In contrast, the right side of Inequality (2) shows the computation costs when T_{child} stays the same position: it is composed of the computational cost to update keys associated with all nodes in T_{child}, with the key sub-path $\mathrm{KSP}_{j,x,r}$ (starting at node x, the root of T_{child} and ending at node r, the root of the key tree. The node j represents a new joining group member which is located at the shallowest leftmost position in the child key tree, T_{child}), and with the key path KP_i (the key path of the leaving group member M_i).

$$N^{Expon.}_{T_{child}+KSP_{j,x,r}} > N^{Expon.}_{T_{child}+KSP_{j,x,r}+KP_i} \quad (2)$$

where N_x^y is the # of y operations for all members in x.

Thus, if moving T_{child} can result in a performance improvement (i.e. Inequality (2) is false), T_{child} should be moved to take $< h, i >$'s position and $< h, i >$ is cut off. This scenario is called *Case 3*.

Otherwise, (i.e. Inequality (2) is true), T_{child} stays at the same position. This is called *Case 4*.

For example, Figure 3 (b) is the original key tree in which the T_{child} is pointed out. Figure 3 (c) shows the key structure change when a group memebr M_2 leaves. Since M_2 is not within T_{child} and moreover, our calculation shows that Inequality (2) is false, T_{child} rooted at $< 2, 2 >$ is moved to node $< 2, 1 >$'s position in order to obtain the performance improvement. The former node $< 2, 1 >$ is cut off. As its left child node is removed, node $< 1, 1 >$ will be deleted. Node $< 1, 1 >$'s right node $< 2, 3 >$ is renamed as $< 1, 1 >$ and it is promoted to its parent's position.

Figure 3 (d) demonstrates that when M_4 leaves, T_{child} need not be moved any other position since M_4 is within T_{child}.

3.2.2 Group Key Updates

To update the group key when a group member leaves, the leave protocol should update all the node keys and blinded keys associated with the nodes in such kind of key paths that have one or more nodes added/deleted. Obviously, the node key and the blinded key of every node within T_{child} should be updated. So do all keys on the leaving member's key path and on the T_{child}'s key path.

Here, we first explain the *dominating key path* concept. Then we describe the proposed *algorithm 1 – dominating algorithm* which updates and forwards the keys on the key sub-paths. At last, we elucidate the *leave protocol*.

Dominating key path: If two key paths intersect, we say that the right key path is dominated by the left key path. Therefore, the left key path is the dominating key path and is responsible for updating the overlapped nodes on the two key paths. For example, in Figure 3 (b), KP_4, the key path for M_4, intersects KP_5, the key path for M_5, at $< 2, 2 >$. Since KP_4 is to the left of KP_5,

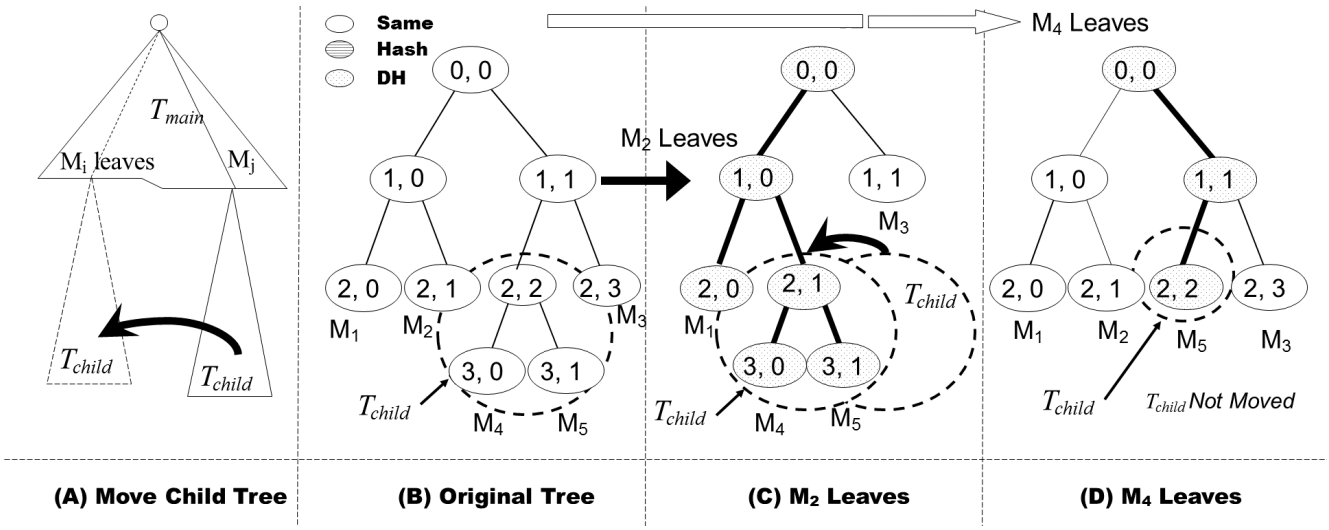


Figure 3: TGDH+: Key tree updates for group members leaving – (a)Move child tree, (b)Original tree, (c) M_2 leaves, and (d) M_5 leaves

Algorithm 1 Dominating Algorithm

1: Begin
2: **for all** sponsor M_i **do**
3: update $KSP_{i,<h,i>,<x_1,y_1>}$
4: **if** all updated blinded keys that associated with key paths which are dominated by M_i already sent out **then**
5: repeat computing *node keys & blinded keys* on its key path until it cannot continue;
6: multicast updated blinded keys on M_i's key path;
7: **else**
8: wait for updated blinded keys associated with key paths which are dominated by M_i;
9: Go to the beginning of step 4;
10: **end if**
11: **end for**
12: **for all** group member M_i **do**
13: update its node keys on its key path after receiving blinded keys from all sponsors.
14: **end for**
15: End

KP_4 dominates KP_5. Therefore, M_4 should update and multicast the blind keys for $< 2, 2 >$.

Algorithm 1 – dominating algorithm: Without consideration for the root of the key tree, assume a key path KP_i intersects $n-1$ other key paths, KP_1, KP_2 ... KP_{i-1}, KP_{i+1}... KP_{n-1}, one by one from the leaf node to the root, where n is an integer and n is less than the height of the tree. Assume that the $n-1$ corresponding intersections are $< x_1, y_1 >, < x_2, y_2 > \ldots < x_{n-1}, y_{n-1} >$. The key path KP_i, is divided into the following n key sub-paths: $KSP_{i,<h,i>,<x_1,y_1>}, KSP_{i,<x_1,y_1>,<x_2,y_2>} \cdots KSP_{i,<x_{n-1},y_{n-1}>,<0,0>}$.

In *dominating algorithm*, a member waits for the updated blinded keys sent from members it dominates. After then, it updates all blinded keys and the node keys on its key path until it cannot. At last, it multicasts all updated blinded keys to other members. Based on these new blinded keys, all group members can update the group key.

For example, in Figure 3 (c), after moving T_{child}, all keys associated with the nodes in T_{child} and T_{child}'s key path are updated:

1^{st} round: Key path of M_5 is dominated by that of M_4. M_5 multicasts $BK_{<3,1>}$.

2^{nd} round: M_4 multicasts $BK_{<3,0>}$, $BK_{<2,1>}$ and $BK_{<1,0>}$. In Figure 4.4 (d), all keys associated with the nodes within T_{child} and T_{child}'s key path are supposed to be updated:

3^{st} round: Key path of M_5 multicasts $BK_{<2,2>}$ and $BK_{<1,1>}$.

Notice that the authentication to secure multicast messages will deploy the digital signing algorithm [24].

Leave Protocol: To update the group key in the case in which a group member leaves, the leave protocol should handle Cases 1, 2, 3, and 4, separately.

Case 1: As showed in Figure 2, the proposed leave protocol is as same as that for TGDH. All auxiliary group keys for every group member are released.

Case 2: As shown in Figure 4, to obtain performance gain, this leave protocol does not update the DH-based keys in the key tree for Case 2 but updates the group key via *Hash* with the auxiliary group key as input. The specific idea behind this proposal is that group members in T_{Main} can be aware of key material which is not known by members in T_{child}. Therefore, after a member which belongs to T_{child}, leaves, the group members in T_{Main} can calculate a new group key which cannot be compromised by the group members in T_{child}, including the leaving one. Then, a designated member in T_{Main} delivers the new group key to a designated member in T_{child} within a secure channel, who, in turn, sends the group key to other

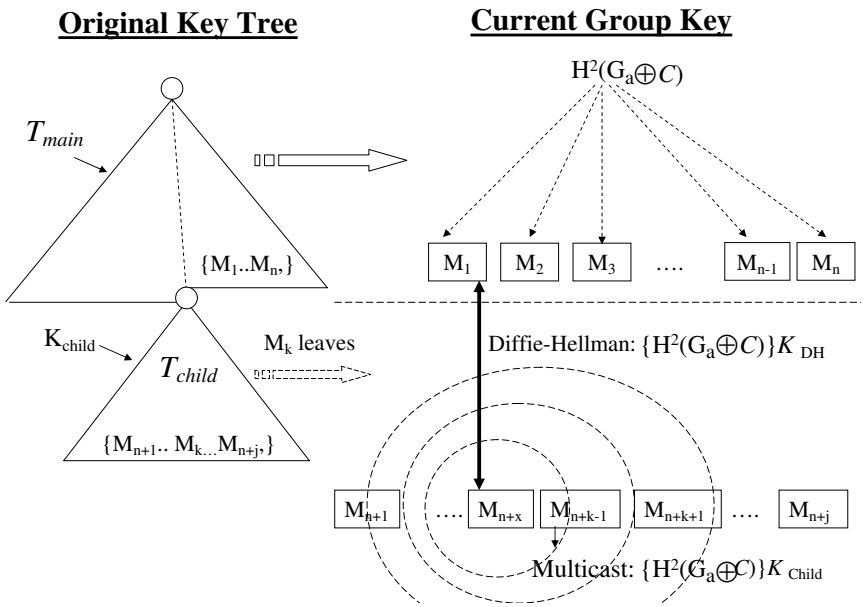


Figure 4: TGDH+: Group key updates for Case 2

members in T_{child} via a secure multicast channel.

The following is a method for calculating the current group key, $G_{current}$, and for updating the auxiliary group key G_a.

Group Key Updates: If group member $M_{n+k} \in \{M_{n+1} \dots M_{n+j}\}$ leaves where $0 < k \leq j$, every group member $\in \{M_1 \dots M_n\}$ can calculate a new group key $G_{current} = H^2(G_a \oplus C)$ based upon its G_a where C is an integer known in advance. As shown in Figure 4, the leftmost leaf of T_{main}, for example, M_1 launches a 2-party DH scheme with a leaf of T_{child}, for example, M_{n+x}, to generate a shared key, which is used to encrypt $G_{current} = H^2(G_a \oplus C)$. Notice that in T_{child} the key path for M_{n+x} is the leftmost updated key path. After using the dominating algorithm to update the keys associated with the nodes in T_{child}, M_{n+x} multicasts the $BKEY_{n+x} \ || \ (G_{current})\ K_{child}$ where K_{child} is the new node key associated with the root of T_{child}. Therefore, every group member in T_{child} can calculate the new sub-group key and decrypt $G_{current}$. Every group member in T_{Main} should update G_a with $H(G_a)$ which can be used to generate future group keys when another group member in T_{child} leaves.

Auxiliary Group Key Updates: After the group key is generated, new auxiliary group keys should be prepared for future group key updates. All members in T_{child} should release the auxiliary group key. All auxiliary group keys G_a stored by members in T_{main} should be replaced by the following formula: $G_a = H(G_a \oplus C)$.

Case 3: As shown in Figure 2 and Figure 3 (c), our protocol should update the DH-based keys associated with T_{child}, the key path of M_i and the key path of M_j via *dominating algorithm.*

Case 4: As shown in Figure 2 and Figure (d), our protocol should update the DH-based keys associated with T_{child}, and the key path of M_i via *dominating algorithm.*

3.2.3 Merge and Partition Protocols

When the group is divided into sub-groups, the partition protocol will treat the members who cannot be in contact with the group as leaving members. In this case, each group member will handle the *0 join & L leave* scenario. In a similar way, when sub-groups merge, the merging protocol deals with the *J join & 0 leave* leave scenario. For every sub-group, the group member hosting the leftmost shallowest key path is treated as the sponsor for the sub-group which generates the new session secret key, updates keys on its key path and multicasts the updated keys. Both the merge protocol and the partition protocol can use algorithm 1: *Dominating Algorithm* to handle the *J join & 0 leave* and *0 join & L leave* scenario respectively.

For example, the procedure to merge 8 sub-groups into a super group is shown in Figure 5. $S_1 \dots$ and S_8 are selected as sponsors for the 8 sub-groups respectively. Using the dominating algorithm, the protocol can generate the group key within 3 rounds.

1^{st} round: The key path for S_2 is dominated by that of S_1. The key path for S_4 is dominated by that of S_3. The key path for S_6 is dominated by that of S_5. The key path for S_8 is dominated by that of S_7. M_2, M_4, M_6 and M_8 update node keys and blinded keys on their key paths, respectively. Then, M_2, M_4, M_6 and M_8 multicast the updated blinded keys on their key sub path starting at the leaf node and ending at $< 3, 1 >, < 3, 3 >, < 3, 5 >$, and $< 3, 7 >$ respectively.

2^{nd} round: The key path for S_3 is dominated by that of S_1. The key path for S_7 is dominated by that of S_5. Then, after calculating these node and blinded keys on their key

Table 2: Computational cost

Scheme	Protocol	Main sponsor			Total		
		Exponen.	H/E^2	Signing	Exponentiation	H/E^2	Signing
TGDH	J j.&1 l.[1]	2h(J+1)	-	J+1	(2n-1)(J+1)	-	2J+1
	Merge	2h	-	Log_2k+1	2(h-$\log_2$k)k+(2k-1)	-	2k
	Partition	2h	-	min($\log_2$p+1,h)	2(h-$\log_2$p)p+(2p-1)	-	min(2h,2p)
STR	J j.&1 l.[1]	4J+(3n/2+2)	-	J+1	(2n+2)J+(3n/2+2)	-	2J+1
	Merge	3m+1	-	2	(n+m)m+3m+1	-	k+1
	Partition	3n/2+2	-	1	(n-1) (3n/4+1) +3n/2+2	-	1
TGDH+	J j.&1 l.[1]	2(h+$\log_2$J)	J+2	1	6J+4n-4	J(J+2n+1)	J/2+1
	Merge	2h	-	1	2(h-$\log_2$k)k+(2k-1)	-	k
	Partition	2h	-	1	2(h-$\log_2$p)p+(2p-1)	-	p

1: J Join & 1 Leave; 2: Hash / Encryption

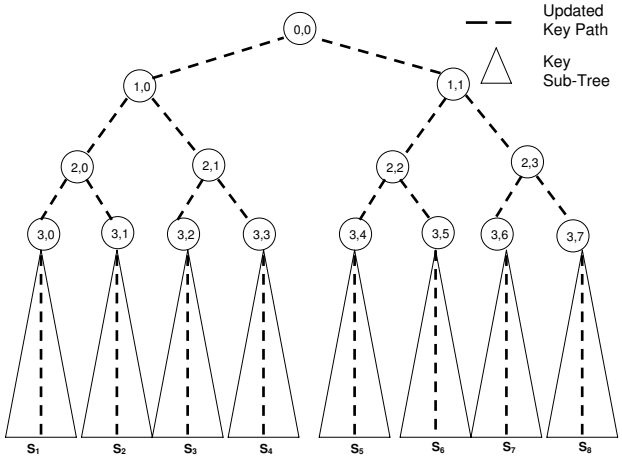


Figure 5: TGDH+: Merge protocol for 8 sub-groups

paths, M_3 and M_7 multicast the updated blinded keys on their key sub path starting at the leaf node and ending at $< 2, 1 >$ and $< 2, 3 >$ respectively.

3^{th} round: M_1 and M_5 update node keys and blinded keys on their key paths, respectively. Then, M_1 and M_5 multicast the updated blinded keys on their key sub path starting at the leaf node and ending at $< 1, 0 >$ and $< 1, 1 >$ respectively.

The partition protocol follows the same procedure. For simplification, the partition protocol will not be introduced again. Furthermore, faults can occur even in join/leave/merge/partition protocols in the contributory group schemes. For joining/merging, the failure node is treated as a leaving member. The paper simply treats them as members who leave. Then it is the leave/partition protocols' turn to handle them. The detailed procedure for leave/partition protocols follows what the leave/partition protocols do: deleting the leaving member's node and its parent node. The leaving node's sibling is promoted to its parent's position. The others functions in the same manner as described earlier.

3.2.4 Authentication and Security Property

Unicasts utilized in this paper can be protected by ID-based Diffie-Hellman key exchange scheme [14] or digital signing algorithms [20]. Multicasts by the Signature Amortization Information Dispersal Algorithm (SAIDA) [30]. The security of TGDH+ is based on the assumptions of 2-party Decision Diffie-Hellman problem (DDH) [24], one way hash function (Hash) [24] and Decision Binary Tree Diffie-Hellman problem (DBTDH) [24]. Please refer to [24] for details. Notice that, as mentioned in [24], the definition of backward and forward secrecy of TGDH is stronger than that of previous group key schemes such as GDH [6]. Our proposal follows the latter. Notice that the authentication to secure multicast messages will deploy the M-SAIDA.

4 Performance Analyses

TGDH [15] and STR [16] have been shown to be among the most efficient contributory group key management schemes. Please refer to [4] for a detailed comparison. We compare our proposal with TGDH and STR. In Tables 2 and 3, we summarize the computational cost and communication overhead of TGDH+, TGDH and STR.

The current group size is denoted by n and the height of the key tree for TGDH and TGDH+ is h. For the merge protocol, the number of sub-groups is k and the number of group members in every sub-group is m. For a partition protocol, the number of leaving members is p. For TGDH and TGDH+, the overhead varies according to the balance of the key tree and the join or leave members location in the key tree. Our performance analysis for them is based on the average scenario. In Tables 2 and 3, both the total cost and the main sponsors cost comprises the cost for all the group members.

J join & 1 leave: As seen from Table 2, TGDH+ is comparatively efficient in terms of the number of exponentiations and the number of signing operations. In Table 3, both STR and TGDH demand the most communication

Table 3: Communication overhead and memory consumption

Scheme	Protocol	Rounds	Communication overhead				Memory
			Main sponsor		Total		
			Unicast	Multicast	Unicast	Multicast	
TGDH	J join&1 leave	2J+1	-	[1, 2J+1]	-	2J+1	0
	Merge	$\log_2$k+1	-	H	-	2k	0
	Partition	min($\log_2$p+1, h)	-	H	-	min(2h,2p)	0
STR	J join&1 leave	2J+1	-	2	-	2J+1	0
	Merge	2	-	1	-	k+1	0
	Partition	1	-	1	-	p	0
TGDH+	J join&1 leave	2J+3	1	1	2J+2	J/2	[0, 1]
	Merge	$\log_2$k+1	-	1	-	k	0
	Partition	min($\log_2$p+1, h)	-	1	-	1	0

overhead. Our scheme requires two more rounds than TGDH and STR. However, the communication scheme deployed for every round is a one-hop unicast. In contrast, the other two schemes use multi-hop multicast for every round which means a larger communication overhead to send the rekey messages around the network. In terms of storage costs, most members of TGDH+ should store one more auxiliary group key than TGDH and STR.

Merge: Our scheme requires less cost as compared to TGDH and STR in terms of the number of multicast messages and computational cost. STR needs the most number of exponentiation operations and TGDH requires the most number of signing operations. STR uses a constant number of rounds.

Partition: TGDH demands the most communication overhead and the most signing operations. STR requires a constant number of rounds, the least numbers of signing operations and the least number of multicast messages. But STR demands the most computational cost, $O(n^2)$ times of exponentiations. So, in terms of computational and communication cost, our scheme is more efficient.

Finally, our TGDH+ is more efficient in *J join & 1 Leave* and merge protocols. For partition protocols, STR works better in signing and multicast metrics. For the rest metrics of the partition protocol, TGDH+ works better. For details of cost comparisons, please refer to Appendix A and B.

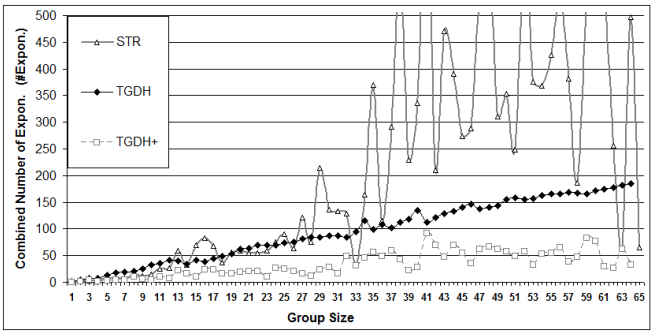


Figure 6: Individual rekey: Number of exponentiations

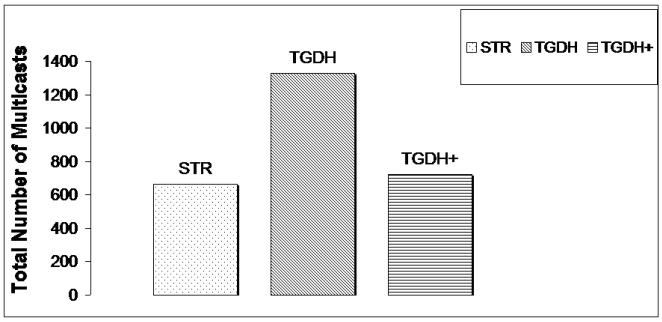


Figure 7: Individual rekey: Total number of multicasts

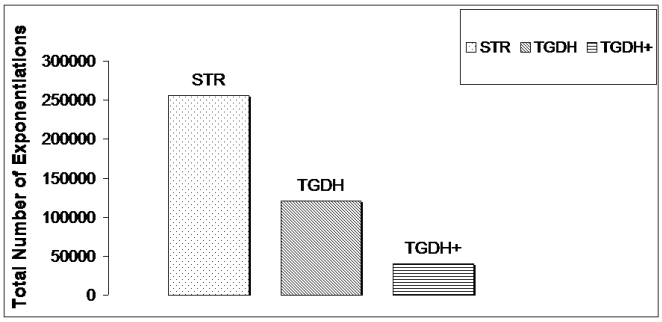


Figure 8: Individual rekey: Total number of exponentiation

5 Experimental Results

Our experiments compare the computational cost and communication for TGDH, STR and TGDH+. It is based on a group membership behavior data set [1] that includes member join time and duration captured on the MBone [2, 3]. In terms of computational cost, the number of exponentiations (hash and Encrypt/Decrypt operations are included via translating them into exponentiation with the ratio of 0.002) for different group sizes is listed in Figure 6. The total number of exponentiations

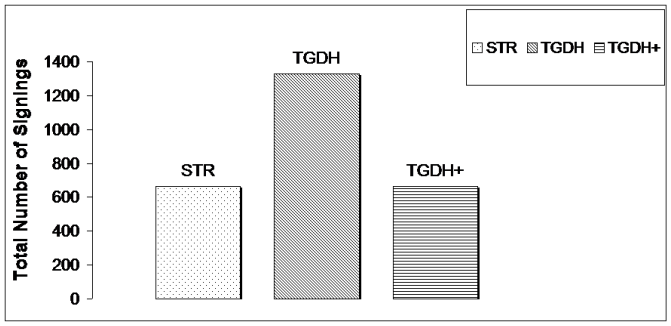


Figure 9: Individual rekey: Total number of signing operations

for every group session is listed in Figure 8. The total number of signings for every group session is listed in Figure 9. With regard to the communication overhead, the total number of multicasts for every group session (unicast is included via translating it into multicast with the ratio of $n^{-0.8}$ where n is a group size) is listed in Figure 7. The results show that our proposal is the most efficient in terms of computational cost. It can be observed that STR requires less number of multicasts than TGDH+. However, the multicast STR used covers the whole group and that of TGDH+ covers only the sub-group.

6 Conclusions

The design of efficient group key management schemes for dynamic peer groups over resource-constrained networks is still a challenging task. This paper presents the design and specification of a lightweight and high performance group key management scheme with the utilization of hash and DH. Performance evaluation and experimental results show that our proposal is more efficient as compared to previously proposed popular contributory group key management schemes.

References

[1] K. C. Almeroth and M. H. Ammar, "Group communication dataset," 2001. (`ftp://ftp.cc.gatech.edu/people/kevin/release-dat`)

[2] K. C. Almeroth, "A long-term analysis of growth and usage patterns in the multicast backbone (MBone)," in *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, vol. 2, pp. 824–833, 2000.

[3] K. C. Almeroth and M. H. Ammar, "Multicast group behavior in the internet's multicast backbone (MBone)," *IEEE Communications Magazine,*, vol. 35, no. 6, pp. 124–129, 1997.

[4] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stanton, and G. Tsudik, "Secure group communication using robust contributory key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 5, pp. 468–480, 2004.

[5] G. Ateniese, M. Steiner, and G. Tsudik, "Authenticated group key agreement and friends," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pp. 17–26, 1998.

[6] G. Ateniese, M. Steiner, and G. Tsudik, "New multiparty authentication services and key agreement protocols," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 628–639, 2000.

[7] D. Boneh, G. Durfee, and M. Franklin, "Lower bounds for multicast message authentication," in *Advances in Cryptology (Eurocrypt'01)*, pp. 437–452, Springer, 2001.

[8] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 255–264, 2001.

[9] Y. Challal and H. Seba, "Group key management protocols: A novel taxonomy," *International Journal of Information Technology*, vol. 2, no. 1, pp. 105–118, 2005.

[10] Q. Cheng, "Security analysis of a pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *International Journal of Network Security*, vol. 17, no. 4, pp. 494–496, 2015.

[11] Q. Cheng and C. 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.

[12] K. Y. Choi, J. Y. Hwang, and D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," in *Public Key Cryptography (PKC'04)*, pp. 130–144, Springer, 2004.

[13] J. C. I. Chuang and M. A. Sirbu, "Pricing multicast communication: A cost-based approach," *Telecommunication Systems*, vol. 17, no. 3, pp. 281–297, 2001.

[14] A. Fekete, N. Lynch, and A. Shvartsman, "Specifying and using a partionable group communication service. Extended version," in *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 53–62, 1997.

[15] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pp. 235–244, 2000.

[16] Y. Kim, A. Perrig, and G. Tsudik, "Group key agreement efficient in communication," *IEEE Transactions on Computers*, vol. 53, no. 7, pp. 905–921, 2004.

[17] A. Kumar and S. Tripathi, "Anonymous ID-based group key agreement protocol without pairing," *International Journal of Network Security*, vol. 18, no. 2, pp. 263–273, 2016.

[18] P. P. Lee, J. Lui, and D. K. Yau, "Distributed collaborative key agreement and authentication protocols for dynamic peer groups," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 263–276, 2006.

[19] D. Li, Z. Aung, S. Sampalli, J. Williams, and A. Sanchez, "Privacy preservation scheme for multicast communications in smart buildings of the smart grid," *Smart Grid and Renewable Energy*, vol. 4, no. 4, pp. 313–324, 2013.

[20] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, "Efficient and fault-diagnosable authentication architecture for ami in smart grid," *Security and Communication Networks*, vol. 8, no. 4, pp. 598–616, 2015.

[21] D. Li and S. Sampalli, "A hybrid group key management protocol for reliable and authenticated rekeying," *International Journal of Network Security*, vol. 6, no. 3, pp. 270–281, 2008.

[22] W. T. Li, C. H. Ling, and M. S. Hwang, "Group rekeying in wireless sensor networks: A survey," *International Journal of Network Security*, vol. 16, no. 6, pp. 401–410, 2014.

[23] M. S. Manasse, "A survey of micropayment technologies, and the millicent system," 1999. (`http://www-db.stanford.edu/infoseminar.Archive/`)

[24] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, CRC press, 1996.

[25] L. E. Moser, Y. Amir, P. M. Melliar-Smith, and D. A. Agarwal, "Extended virtual synchrony," in *Proceedings of the 14th IEEE International Conference on Distributed Computing Systems*, pp. 56–65, 1994.

[26] M. J. Moyer, J. R. Rao, and P. Rohatgi, "A survey of security issues in multicast communications," *IEEE Network*, vol. 13, no. 6, pp. 12–23, 1999.

[27] V. S. Naresh and N. V. Murthy, "Elliptic curve based dynamic contributory group key agreement protocol for secure group communication over Ad-hoc networks," *International Journal of Network Security*, vol. 17, no. 5, pp. 588–596, 2015.

[28] N. Okabe, S. Sakane, K. Miyazawa, A. Inoue, M. Ishiyama, and K. Kamada, "A study of security architecture for control networks over IP," in *Ist International Workshop on Networked Sensing Systems (INSS'04)*, 2004.

[29] E. Okamoto and K. Tanaka, "Key distribution system based on identification information," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 481–485, 1989.

[30] J. M. Park, E. K. Chong, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Transactions on Information and System Security*, vol. 6, no. 2, pp. 258–285, 2003.

[31] A. Penrig, D. Song, and J. Tygar, "Elk, a new protocol for efficient large-group key distribution," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 247–262, 2001.

[32] R. S. Ranjani, D. L. Bhaskari, and P. Avadhani, "An extended identity based authenticated asymmetric group key agreement protocol," *International Journal of Network Security*, vol. 17, no. 5, pp. 510–516, 2015.

[33] K. H. Rhee, Y. H. Park, and G. Tsudik, "A group key management architecture for mobile Ad-hoc wireless networks," *Journal of Information Science and Engineering*, vol. 21, no. 2, pp. 415–428, 2005.

[34] S. Setia, S. Koussih, S. Jajodia, and E. Harder, "Kronos: A scalable group re-keying approach for secure multicast," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 215–228, 2000.

[35] A. T. Sherman, D. McGrew, et al., "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.

[36] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, 2000.

[37] Y. Sun and K. Liu, "Securing dynamic membership information in multicast communications," in *(INFOCOM'04). Twenty-third IEEE AnnualJoint Conference on Computer and Communications Societies*, vol. 2, pp. 1307–1317, 2004.

[38] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The versakey framework: Versatile group key management," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 9, pp. 1614–1631, 1999.

[39] H. Weatherspoon, C. Wells, P. R. Eaton, B. Y. Zhao, and J. D. Kubiatowicz, *Silverback: A global-scale archival system*, Computer Science Division, University of California, 2001.

[40] F. Wei, Y. Wei, and C. Ma, "Attack on an ID-based authenticated group key exchange protocol with identifying malicious participants," *International Journal of Network Security*, vol. 18, no. 2, pp. 393–396, 2016.

[41] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

[42] C. K. Wong and S. S. Lam, "Digital signatures for flows and multicasts," in *Proceedings of Sixth IEEE International Conference on Network Protocols*, pp. 198–209, 1998.

[43] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, vol. 1, pp. 345–352, 1999.

[44] W. H. Yang and S. P. Shieh, "Secure key agreement for group communications," *International Journal of Network Management*, vol. 11, no. 6, pp. 365–374, 2001.

[45] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam, "Reliable group rekeying: a performance analysis,"

in *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 27–38, 2001.
[46] X. B. Zhang, S. S. Lam, D. Y. Lee, and Y. R. Yang, "Protocol design for scalable and reliable group rekeying," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 908–922, 2003.
[47] X. Zou, B. Ramamurthy, and S. S. Magliveras, *Secure group communications over data networks*, Springer Science & Business Media, 2007.

Appendices

In this section, we analyze performance-relevant criteria, namely, computational cost and communication overhead for TGDH+. The memory consumption for TGDH+ is already analyzed in Table 3.

A Metrics for Performance Evaluation

A.1 Computational Cost

Every group key scheme comprises a variety of cryptographic operations. To begin with, this paper considers the performance evaluation for each operation. Then, the performance costs for each operation are accumulated to attain the total costs. Previous experiments [28, 38, 23] demonstrate that each cryptographic scheme needs to be processed within a certain period of time, which can be viewed roughly as the performance cost it demands compared with other schemes. Therefore, like other research [23, 28, 38] this paper assumes that the performances of these cryptographic operations can be measured by timing. The experimental results referred to in this paper are listed below.

An experiment result: for the SUN ultra 1/170 workstation, the processing timings for the hash, encryption/decryption, DH, digital signing and digital signing verification operations are 0.01ms, 0.01ms, 100ms, 200ms and 50ms respectively [38], if the key size is 1024 bits.

In [28], for a low-end 8 bits CPU such as H8/3048 or CDS 80390, processing timings for hash, encryption/decryption, and DH operations are 400ms, 400ms, and 400s respectively while key size is 1024 bits. In [23], similar timings have been determined, similar timings have been determined.

According to the results, the hash and encryption/decryption operations show an almost equivalent performance and both of them are about 0.001 times equivalent to a DH operation. Then, insight analyses demonstrate us that every DH key scheme comprises two exponential operations for every party. Therefore, the computational cost for the hash or encryption/decryption operation is 0.002 times that of an exponential operation. So, the number of exponential operations can be treated as the metric when comparing the computational cost of each group scheme which includes different cryptographic operations. The number of encryption/decryption and hash operations can be transferred into the number of exponential operations by a factor of 0.002.

A.2 Communication Overhead:

The areas for evaluating communication overhead consist of the number of rounds, the number of unicasts and the number of multicasts. Previous research [7, 13] shows that the impact of unicasts and multicasts on network bandwidth can be compared with respect to quantification. The costing function shown below was deployed by Chuang and Sirb [13].

$$R_{u/m} = \frac{L_u}{L_m} = n^{-0.8} \tag{3}$$

where n: group size; L_u: average unicast hops; L_m: total hops of a multicast tree;

This research uses it to evaluate the communication overhead between unicasts and multicasts. Utilizing Formula (3), the number of unicasts can be transferred into the number of multicasts and finally each group key scheme is analyzed by comparing the number of multicasts it demands. Therefore, the number of the multicast is the metric for communication overhead for every group key scheme.

A.3 Memory Consumption

In this paper, for the sake of fairness, the key length for every group/auxiliary key should be the same. So, the metric for evaluating memory consumption is the number of group/auxiliary keys stored by every group member.

B Performance Evaluation for Each Group Key Scheme

In this subsection, this paper first introduces the view of group membership events so that subsequent discussion is based upon the same event. Then, the notions of computational costs and communication overhead for the event are defined. Finally, the performances of TGDH+ are discussed.

B.1 Group Session Model

First, let us take a look at the procedures for a group session. Every group session can be treated as a sequence of group members joining and group members leaving. Therefore, this paper assumes that every group session is comprised of a set of *J Join & 1 Leave* ($J \geq 0$) events.

The performance for the *J Join & 1 Leave* ($J \geq 0$) scenarios, which are shown in Figure 10 (group member join/leave for TGDH+), is discussed. In Figure 10 both

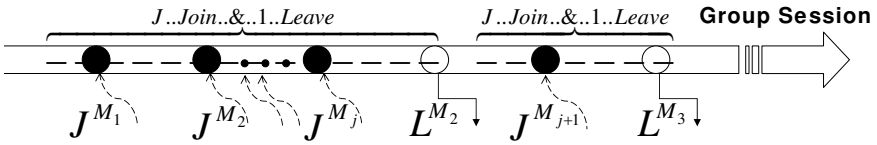


Figure 10: Group session model

the key tree, T_{Main}, and child key tree, T_{child}, are available. Assume that the number of members in T_{main} is n and the number of members in T_{child} is J. For the sake of simplification, assume that $n = 2^x$ and $J = 2^y$ where x and y are integers. Hence, both key trees are balanced.

B.1.1 Computational Cost

Let *COMP(J, n)* denote the combined computational cost for all group members to update the group keys for one *J Join & 1 Leave* event. *COMP(J, n)* is comprised of the number of hash operations, the number of encryption/decryption operations, the number of DH operations and the number of digital signing operations.

$$COMP(J,n) = N_{J,n}^{SIGN} + N_{J,n}^{DH} + N_{J,n}^{ENC} + N_{J,n}^{Hash}$$

where $N_{J,n}^{ENC}$: number of encryption;

$N_{J,n}^{Hash}$: number of Hash;

$N_{J,n}^{DH}$: number of Diffie-Hellman;

$N_{J,n}^{SIGN}$: number of digital signing.

B.1.2 Communication Overhead

Let *COMM(J, n)* denote the combined communication overhead for all group members to update the group keys for one *J Join & 1 Leave* event in which the original group size is n. *COMM(J, n)* is comprised of the number of unicasts and the number of multicasts.

$$COMM(J,n) = N_{J,n}^{Unicast} + N_{J,n}^{Multicast}$$

where $N_{J,n}^{Unicast}$: Number of Unicast;

$N_{J,n}^{Multicast}$: Number of Multicast;

B.2 TGDH+

The *J Join & 1 Leave* scenario, as shown in Figure 10 is analyzed below.

B.2.1 Computational Cost for TGDH+

For every group member joining, every member should use hash to update its group key and the sponsor should encrypt its hash result and send it to the new member. In the case where a group member joins, the join protocol demands DH, Hash and Encryption/Decryption operations. The join protocol for handling J joining requires 2J times the DH operations.

$$N_{J,n}^{Hash} = \sum_{i=1}^{J}(n+i-1) = J(2n+J-1)/2$$

$$N_{J,n}^{ENC} = \sum_{i=1}^{J} 2 = 2J; \qquad N_{J,n}^{DH} = \sum_{i=1}^{J} 2 = 2J$$

When one group member leaves, there are four cases, as discussed earlier.

Case 1: TGDH is used to handle this 0 join & 1 leave scenario.

$$N_{J,n}^{DH} = 2n-1; \qquad N_{J,n}^{SIGN} = 1$$

Case 2: Group members in T_{main} should process hash operations. One DH is launched between a group member in T_{main} and a group member in T_{child}. One encryption and one decryption is also needed between them. In T_{child}, keys associated with the leaves on T_{child} are already computed in the case of the join protocol. All other DH-based keys should be updated and all group members should decrypt the new group key.

$$N_{J,n}^{SIGN} = J/2; \qquad N_{J,n}^{DH} = 3J/2+1;$$

$$N_{J,n}^{ENC} = J+2; \quad N_{J,n}^{Hash} = 2(n-J)$$

Case 3 or Case 4: The leave protocol should update the keys on T_{child} and those on the key path for M_k. Keys associated with the leaves on T_{child} are already computed in the case of the join protocol. So the number of keys to be updated by all members in T_{child} is (J-1). The number of keys to be updated by all members in T_{main} should be 2n-1 due to the updating of M_k's key path.

$$N_{J,n}^{DH} = (J-1) + 2n - 1 + 2J = 3J + 2n - 2$$

B.2.2 Communication Cost for TGDH+

In the case where one group member joins, this proposal's join protocol uses the ID-based Diffie-Hellman authentication which sends two unicast messages to generate the shared key between the sponsor and the new group member. In the case where one group member leaves, according to the *Dominating* algorithm, the number of signing

operations to update T_{child} and the of M_k key path should be $J/2$.

$$N_{J,n}^{Unicast} = 2J; \quad N_{J,n}^{Multicast} = J/2$$

When one group member leaves, there are 4 cases.

Case 1: $N_{J,n}^{Multicast} = 1$

Case 2: $N_{J,n}^{Unicast} = 2$; $N_{J,n}^{Multicast} = J/2$

Case 3 or Case 4: According to the dominating algorithm, the number of signing operations to update for updating T_{child} and M_k's key path should be J/2. This means that $N_{J,n}^{Multicast} = J/2$.

Depeng Li received his Ph.D. degree in computer science from Dalhousie University, Canada in 2010. He has joined Department of Information and Computer Sciences (ICS) at University of Hawaii at Manoa (UHM) as an assistant professor since 2013. His research interests are in security, privacy, and applied cryptography. His research projects span across areas such as Internet of Things, air traffic management, smart grids, and mHealth.

Srinivas Sampalli is a Professor and 3M Teaching Fellow in the Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada. His research interests are in the areas of security and quality of service in wireless and wireline networks. Specifically, he has been involved in research projects on protocol vulnerabilities, security best practices, risk mitigation and analysis, and the design of secure networks. He is currently the principal investigator for the Wireless Security project sponsored by Industry Canada. Dr. Sampalli has received many teaching awards including the 3M Teaching Fellowship, Canada prestigious national teaching award.

An ID-based Hierarchical Access Control Scheme with Constant Size Public Parameter

Rang Zhou, Chunxiang Xu, Wanpeng Li, Jining Zhao
(Corresponding author: Rang Zhou)

Department of Computer Science and Engineering, University of Electronic Science and Technology of China
No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu, 611731, P.R. China
(Email: zhour1987@sohu.com)

(Received Mar. 26, 2013; revised and accepted Jan. 15 & July 13, 2015)

Abstract

Many ways are proposed to reduce the secret storage space of access control in hierarchy, but no one optimizes the public parameters which are only modified by CA. Length of each public parameter is one important factor for the size and utilization of storage space. The frequent changes on the maximum length of public parameter will be a weakness for stability, in dynamic key management. Number and length of changed public parameter are considered for the interaction between CA and trusted public platform in dynamic management. The paper proposes an improved scheme to optimize storage space of public parameter for each class from variable linear size to a small constant size. Our scheme has higher utilization, stability and efficiency on storage space of trusted public platform and needs less interaction between CA and trusted public platform. The security of this improved scheme is proved on key recovery model.

Keywords: Hierarchical access control, ID-based cryptographic, key management, public parameters

1 Introduction

The technology of access control, which is one of the most important components in computer system, is a central issue in computer research. With the great development of information technology, to improve the efficiency of access control and protect the data confidentiality, security access control technology becomes the main research objective recently [5].

To achieve the goal of classifying users and information resource, researchers proposed multi-level security access control model [14]. Each two users are linked by the relationship of binary partial order. The users in high security layer can access the secretly information possessed by low layer users. It is infeasible on the contrary.

To reduce the key storage, Akl and Taylor [1] firstly proposed an approach that the public key cryptography is used on multi-level security access control, in 1983. The approach optimizes key management greatly, which separates key assignment into a public key cryptosystem for the management of all classes' privilege and a symmetric cryptosystem for data protection.

The actual scene for hierarchical access control: In the past, hierarchical access control schemes always are assumed that trusted public platform [4, 8] is a simple storage space only for users' corruption and free for whole system. In fact, we have to admit that the trusted public platform is provided by money. CA buys the trusted public platform to store lots of public parameters, so the money for the management of public parameter is related with the whole system space for public parameter. So, it is necessary to reduce the scale of public parameters of the whole hierarchical access control scheme. Further, we can consider that the trusted public platform has the computation ability to reduce the computation which is done by CA and users as the model of cloud computing.

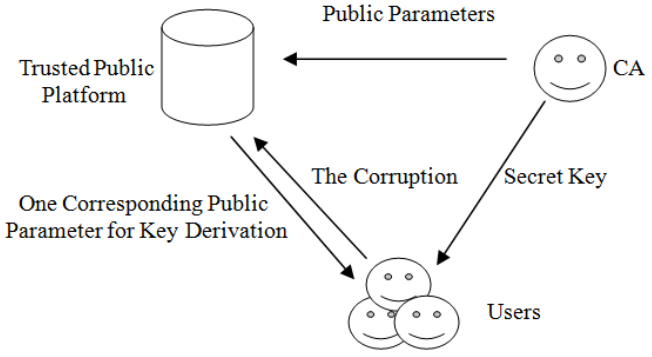


Figure 1: An example of using the proposed scheme

Now, we consider the scene where the whole system is constructed by CA, trusted public platform and users as Figure 1. The trusted platform can provide the computation and storage service [25]. It is necessary to assume that the trusted platform for public parameter is as

honest and curious as the cloud service provider in the model of cloud computing. When a high layer user wants to derive the low layer key, the trusted platform sends only one corresponding public parameter which is used for key derivation simply to the authorized user. Then, the authorized user computes the low layer key by using its secret key and the corresponding public parameter.

Related work: In Akl-Taylor's scheme [1], a CA is needed and if two security classes have a partial order relationship, the low security class's key can be derived from the high security class's key. Though the scheme is simple to understand, but two problems are inevitable in application. The first problem is the storage for the great amount of integers and the other is the computation load that all system needs to update in every change of authorized relationship. To reduce the storage of integers, Mackinnon [21] proposed a typical and improved key allocation scheme. Unfortunately, a lot of integers are needed in this scheme.

Harn and Lin [12] proposed a hierarchical encryption access control scheme based on the hardness of large integer factorization. In contrast with Akl-Taylor's scheme, Harn-Lin's scheme generates key from the low layer to high layer. Though the scheme reduces the time of public parameters generation, the number of integers is same to Akl-Taylor's scheme.

Combining with the security assumptions of Akl-Taylor's scheme and Harn-Lin's scheme, Hwang-Yang [15] proposed an efficient hierarchical access control scheme. The scheme effectively reduces the number of integers, but it is not secure under collusion attack [28].

A YCN scheme are proposed to solve the key assignment problem with a matrix model by Yeh et al. [30]. To protect the data security, in their scheme, two key are used which are a derivation key and an encrypted data key. However, Hwang shown that several user classes can collaborate to derive the derivation keys and encryption keys in some cases under YCN scheme [13, 17]. In 2003, to fix the collusive attack problem, Yeh et al. and Lin proposed their improved schemes, respectively [17, 31].

To solve the problem of key security, Tzeng proposed a time-bound cryptographic key assignment scheme in a partially ordered hierarchy in 2002 [27]. Each authorized user can access the specified data by the legal key during the authorized period only. However, Yi and Ye point out the insecurity of Tzeng's scheme, that the secret keys of some classes can be derived by any three users' collaboration [32]. To solve the problem, Chien and Santis proposed their schemes about time-bound cryptographic key assignment scheme in a partially ordered hierarchy [6, 24].

Lo [20] proposed a new efficient hybrid key assignment scheme in 2011. The security of Lo's scheme is based on the hardness of one-way function and large integer factorization. The scheme is proved more efficiently than the previous schemes with same type. However, two efficiency problems of Lo's scheme must be considered on the actual scene. They are described as following: 1). To resist the partial key exposure attack, every e_i satisfies the condition $m^{7/8} \leq e_i \leq m$ [3]. The public parameters of high layer class are constructed by the e_i's product of all child nodes. Obviously, the length of public parameter from low layer class to high layer class gradually increases. So, the public parameter of high class will be large. It is bad for the cost on buying the space of public platform. 2). In the dynamic key management phase, every modification about class adding or deleting corresponds to a great mount of modification on public parameters.

Nikooghadam [22], Wu [29], and Odelu [23] proposed their schemes which are constructed on ECC. Lo [19] and Lee [16] proposed their schemes Based on Polynomial. These schemes have high efficiency on key derivation and key management, but each authorized relationship needs a public parameter to be stored on the public trusted platform. The amount of public parameters in these schemes largely surpasses the number of the secure classes. These schemes not only don't take full advantage of hierarchy access control, but increase the cost of public parameters storage and maintenance.

Many other schemes have been proposed to solve the problem of access control in a hierarchy [9, 10, 11, 26]. However, two weaknesses are inevitably in these independent key management schemes. The first one is path searching from starting class to target class. The other is the information interaction [33], which contains all public parameters on the path, between the public trusted platform and users.

Our contribution: The contributions of this paper can be summarized as follows:

1) Firstly, to resolve the two problems proposed in Lo's scheme, the paper proposes an improved scheme with constant size of public parameters. The improved scheme optimizes every public parameter, which stores on trusted public platform, to constant size. So, comparing with Lo's scheme, the improved scheme has the advantage as following: To reduce the storage of public parameters greatly; to improve the storage utilization of public parameters greatly; to decrease the number of modification on public parameters in dynamic key management; to strengthen the system stability. At the same time, the optimization of public parameters makes the key derivation process more efficiently than Lo's scheme.

2) Combining with Lo's scheme, the paper introduces the security classes' ID to a part of public key and proposes an ID-based hierarchical access control scheme. By changing every public parameter to a simple random value in constant size, the scheme reduces the size of storage for public parameter smaller. As a result, the scheme optimizes the storage space and utilization of public parameter further.

Roadmap: The remainder of the paper is structured as follows. In Section 2, the preliminaries for security analysis are introduced. Section 3 presents the new improved scheme including key generation, key derivation

and dynamic key management. The performance comparing with the original scheme is provided in Section 4. The security proof of the new improved scheme is discussed in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 The Key Recovery Attack Model for Hierarchical Access Control Scheme

Definition 1. *(Key Recovery Model) [2]. A key allocation scheme is secure in key recovery if no polynomial time adversary A has a non-negligible advantage (in the security parameter ρ) against the challenger in the following game:*

- *Setup: The challenger C runs Setup(1^ρ,G), and gives the resulting public information Pub to the adversary A.*
- *Attack: The adversary issues, in any adaptively chosen order, a polynomial number of Corrupt(v_i) queries, which the challenger C answers by retrieving $k_i \leftarrow Sec(v_i)$ and giving k_i to A.*
- *Break: The adversary outputs a node v^*, subject to $v^* \notin Des(v_i, G)$ for any v_i asked in Attack Phase, along with her best guess k'_{v^*} to the cryptographic key k_{v^*} associated with node v^*.*

We define the adversary's advantage in attacking the scheme as: $Adv_A^{REC} = Pr[k'_{v^*} = k_{v^*}]$.

The Corrupt(v_i), which is proposed by adversary A, is equivalent to collusion attack within the system. The union of adversary's keys is same as the collusive behavior which the low security classes do. They have the same object to get the high security classes' keys. So, it is the static security model in hierarchical system.

2.2 Hard Problems and Assumption

Definition 2. Discrete log-problem: *G is a finite cyclic group, whose generator is g and has order of $n = |G|$, where it is easy to compute $g^a = A$ from a and hard to compute $a', 0 \le a' \le n$ which satisfies the condition $g^{a'}$, from A.*

Definition 3. Strongly Hash function:

1) The input of $H(.)$ is no any restriction on length,

2) The output of $H(.)$ is constant length and can resist the birthday attack,

3) It is simple to compute the value of $H(x)$ from the known x, but it is computationally infeasible on the contrary,

4) No one can feasibly finds two different values of x that give the same $H(x)$.

Definition 4. Large integer factorization problem: *For the given odd composite number N constructed by two prime factor, it is hard to compute the prime p which satisfies the condition $p|N$ in appropriate environment.*

Definition 5. RSA problem: *The number $N = p \cdot q$ is known, where the number p and q both are primes. The number e is an integer and satisfies the condition $\gcd(e, (p-1)\cdot(q-1)) = 1$. It is infeasible to compute the unique integer $m \in Z_n^*$, where m satisfies the condition $m^e = c(modN)$ from the fixed $c \in Z_n^*$.*

3 The Proposed Scheme

The paper proposes an improved ID-based hierarchical access control scheme with constant size public parameter. The scheme is comprised of key generation, key derivation and dynamic key management.

3.1 Key Generation Phase

Firstly, a CA is needed to do the work of key computation and assignment for the authorized users. CA executes the following steps:

Step 1. CA chooses two large primes p and q, and computes the public large number $m = p \cdot q$ and secret parameter $\varphi(m) = (p-1)\cdot(q-1)$, where $\varphi(m)$ must be kept secretly by CA and m is kept on the trusted public platform. At last, CA destroys p and q.

Step 2. CA generates a random number g, which is coprime with m and $2 < g < (m-1)$. CA chooses two public one-way hash function $H_1(.)$, $H_2(.)$ and $m^{7/8} \le H_1(.) \le m$, $\gcd(H_1(.), \varphi(m)) = 1$, $H_2(.) \le m$ [3].

Step 3. In the hierarchical access control, each class C_i has an ID_i.

Step 4. For every class C_i, which is a non-leaf class or a leaf class with two or more immediate ancestors in the hierarchy, CA computes $e_i = H_1(ID_i)$. Then, CA computes private key exponent $d_i = e_i^{-1} mod \varphi(m)$. The pair (e_i, d_i) is corresponding to C_i, where the secret key d_i is only kept secretly by CA.

Step 5. Key generation.

- For every class C_i, which is a non-leaf class or a leaf class with two or more immediate ancestors in the hierarchy, CA computes the secret key $K_i = g^{d_i \prod_{all C_l} d_l mod \varphi(m)} \bmod m$, where the all C_l is the successor of C_i and no one is a leaf class with only one immediate ancestor.
- For every leaf class C_i, which has only one immediate ancestor C_j in the hierarchy, CA randomly generates a secret key K_i for the corresponding class C_i and calculates a public parameter $PB_i = K_i \oplus H_2(K_j, ID_i, ID_j)$.

Step 6. CA passes every secret key K_i to corresponding class C_i through a secure channel individually and publishes all public parameters and authorized relationships on public platform of trusted.

For a clear description for the initialization and key generation, an example of the proposed scheme, where the classes have the authorized relationships, is shown in Figure 2. The prime pairs, secret keys and public parameters about dependent key are generated as Table 1. The secret keys and public parameters about independent key are generated as Table 2.

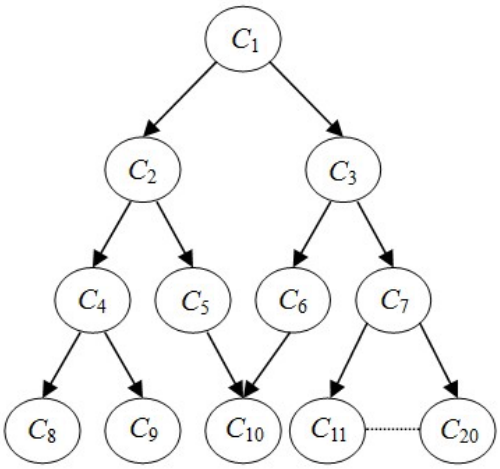


Figure 2: An example of using the proposed scheme

Table 1: The dependent key in the example

Class (C_i)	Secret key (K_i)	Prime pair (e_i, d_i)
C_1	$g^{d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_{10} mod\varphi(m)} \bmod m$	(e_1, d_1)
C_2	$g^{d_2 d_4 d_5 d_{10} mod\varphi(m)} \bmod m$	(e_2, d_2)
C_3	$g^{d_3 d_6 d_7 d_{10} mod\varphi(m)} \bmod m$	(e_3, d_3)
C_4	$g^{d_4 mod\varphi(m)} \bmod m$	(e_4, d_4)
C_5	$g^{d_5 d_{10} mod\varphi(m)} \bmod m$	(e_5, d_5)
C_6	$g^{d_6 d_{10} mod\varphi(m)} \bmod m$	(e_6, d_6)
C_7	$g^{d_7 mod\varphi(m)} \bmod m$	(e_7, d_7)
C_{10}	$g^{d_{10} mod\varphi(m)} \bmod m$	(e_{10}, d_{10})

3.2 Key Derivation Phase

3.2.1 The Formula of Key Derivation

C_i and C_j are in the hierarchy with relationship $C_i \preceq C_j$. Secret key K_i, K_j are corresponding to classes C_i, C_j, respectively. The formula of key derivation from K_j to K_i is used as following:

Table 2: The independent key in the example

Class(C_i)	Public parameter(PB_i)	Secret key(K_i)
C_8	$K_8 \oplus H_2(K_4, ID_8, ID_4)$	K_8
C_9	$K_9 \oplus H_2(K_4, ID_9, ID_4)$	K_9
C_{11}	$K_{11} \oplus H_2(K_7, ID_{11}, ID_7)$	K_{11}
$\cdots$	$\cdots$	$\cdots$
C_{20}	$K_{20} \oplus H_2(K_7, ID_{20}, ID_7)$	K_{20}

1) C_i is a leaf class with only one immediate ancestor C_j, $K_i = PB_i \oplus H_2(K_j, ID_i, ID_j)$.

2) If C_i is a leaf class with only one immediate ancestor C_k, $K_i = PB_i \oplus H_2(K_j^{\prod_{allC_l, C_l \preceq C_j \wedge C_l \npreceq C_k} H_1(ID_l)}, ID_i, ID_k)$.

3) Otherwise, $K_i = K_j^{\prod_{allC_l, C_l \preceq C_j \wedge C_l \npreceq C_i} H_1(ID_l)}$.

3.2.2 Correctness Proof

Theorem 1. *For two classes C_i and C_j with the relationship $C_i \preceq C_j$, C_j can derive the secret key K_i of C_i from the above formula.*

Proof. In a hierarchical access control system with n nodes, we have the public parameters $PB_1, PB_2, ...PB_n$, the one-way hash function $H_1(.)$, $H_2(.)$ and $m^{7/8} \leq H_1(.) \leq m$, $H_1(.) \in prime$, $gcd(H_1(.)\varphi(m)) = 1$, $H_2(.) \leq m$.

1) Class C_i is a leaf node with only one immediate ancestor C_j. It is simple to get the key from the equation $K_i = PB_i \oplus H_2(K_j, ID_i, ID_j)$.

2) Class C_i is a leaf node with only one immediate ancestor C_k and the relationship of $C_i \preceq C_k \preceq C_j$ is satisfied, where C_k is the immediate ancestor of C_i. Because we do not know the secret key K_k, we shall process the following steps. Every C_l is the successor of C_j, except the leaf class with only one immediate ancestor.

$$\begin{aligned} K_i &= PB_i \oplus H_2(K_k, ID_i, ID_k) \\ &= PB_i \oplus H_2(g^{d_k \prod_{allC_l, C_l \preceq C_k} d_l} \bmod m, \\ &\qquad ID_i, ID_k) \\ &= PB_i \oplus H_2(K_j^{(\frac{\prod_{allC_l, C_l \preceq C_j} e_l}{\prod_{allC_l, C_l \preceq C_k} e_l})} \bmod m, \\ &\qquad ID_i, ID_k) \\ &= PB_i \oplus H_2(K_j^{\prod_{allC_l, C_l \preceq C_j \wedge C_l \npreceq C_k} H_1(ID_l)} \bmod m, \\ &\qquad ID_i, ID_k). \end{aligned}$$

3) Class C_i is a non-leaf class or a leaf class with two or

more immediate ancestors in the hierarchy.

$$\begin{aligned} K_i &= g^{d_i \prod_{allC_l, C_l \preceq C_i} d_l} \mod m \\ &= K_j^{(\prod_{allC_l, C_l \preceq C_j} e_l / (\prod_{allC_l, C_l \preceq C_i} e_l))} \mod m \\ &= K_j^{\prod_{allC_l, C_l \preceq C_j \wedge C_l \not\preceq C_i} H_1(ID_l)} \mod m \end{aligned}$$

□

3.2.3 Example of Key Derivation

To describe the key derivation clearly, the example as Figure 2 is used.

1) The key derivation from C_4 to C_9:

$$K_9 = PB_9 \oplus H_2(K_4, ID_9, ID_4).$$

2) Because of the relationship of $C_{11} \preceq C_7 \preceq C_1$ in the hierarchy, the key derivation from C_1 to C_{11} is

$$K_{11} = PB_{11} \oplus H_2(K_1^{\prod_{allC_l, C_l \preceq C_1 \wedge C_l \not\preceq C_7} H_1(ID_l)} \mod m, ID_{11}, ID_7).$$

The proof is stated as follows:

$$\begin{aligned} K_{11} &= PB_{11} \oplus H_2(K_7, ID_{11}, ID_7) \\ &= PB_{11} \oplus H_2(g^{d_7} \mod m, ID_{11}, ID_7) \\ &= PB_{11} \oplus H_2(g^{(d_7 d_1 d_2 d_3 d_4 d_5 d_6 d_{10})(e_1 e_2 e_3 e_4 e_5 e_6 e_{10})} \mod m, ID_{11}, ID_7) \\ &= PB_{11} \oplus H_2(K_1^{\prod_{allC_l, C_l \preceq C_1 \wedge C_l \not\preceq C_7} H_1(ID_l)} \mod m, ID_{11}, ID_7). \end{aligned}$$

3) The key derivation from C_1 to C_4 is the formula

$$K_4 = K_1^{\prod_{allC_l, C_l \preceq C_1 \wedge C_l \not\preceq C_4} H_1(ID_l)} \mod m.$$

The proof is stated as follows:

$$\begin{aligned} K_4 &= g^{d_4} \mod m \\ &= g^{(d_4 d_1 d_2 d_3 d_5 d_6 d_7 d_{10})(e_1 e_2 e_3 e_5 e_6 e_7 e_{10})} \mod m \\ &= K_1^{\prod_{allC_l, C_l \preceq C_1 \wedge C_l \not\preceq C_4} H_1(ID_l)} \mod m. \end{aligned}$$

3.3 Dynamic Key Management

It is necessary to provide the dynamic key management ability for a hierarchical key assignment scheme. This section discusses the key changes of adding and deleting classes.

3.3.1 Adding a New Class with ID_{new}

1) Adding a New Leaf Class with Only One Immediate Ancestor:

   a. The immediate ancestor of the new class is not a leaf class with only one immediate ancestor before the adding.

   The new class C_{new} is the immediate successor of C_r. CA generates random key K_{new} and XOR public parameter PB_{new} as Step 5 in key generation phase. At last, CA modifies the authorized relationship to add C_{new}.

   b. The immediate ancestor of the new class is a leaf class with only one immediate ancestor before the adding.

   The new class C_{new} is an immediate successor of C_r. CA generates pair (e_r, d_r) as Step 4, and new key K_r and random key K_{new} for C_r and C_{new} as Step 5 in key generation phase, respectively. Then, CA modifies the secret keys of C_r's ancestors, and the public parameters of C_r and C_{new}. At last CA modifies the authorized relationship to add C_{new}.

2) Adding a New Non-leaf Class with Only One Immediate Ancestor or a Class with Multiple Immediate Ancestors in the Hierarchy:

   a. The new class is a non-leaf class with only one immediate ancestor.

   The new class C_{new} is the immediate successor of C_{r_1} and the immediate ancestor of C_{r_2}. CA generates pair (e_{new}, d_{new}) as Step 4, and new key K_{new} as Step 5 in key generation phase. Then, CA modifies the secret keys of C_{new}'s ancestors. At last CA modifies the authorized relationship to add C_{new}.

   b. The new class is a class with multiple immediate ancestors in the hierarchy.

   The new class C_{new} is the immediate successor of C_{r_1} and C_{r_2}. CA generates pair (e_{new}, d_{new}) as Step 4, and new key K_{new} as Step 5 in key generation phase. If C_{r_1} is a leaf class with only one immediate ancestor, CA generates pair (e_{r_1}, d_{r_1}) as Step 4. If C_{r_2} is a leaf class with only one immediate ancestor, CA does the same work for C_{r_2} as the scene of C_{r_1}. Then, CA modifies the secret keys of C_{new}'s ancestors. At last CA modifies the authorized relationship to add C_{new}.

3.3.2 Removing a Class with ID_{del}

1) Removing a Leaf Class with Only One Immediate Ancestor:

   a. The immediate ancestor of the removing class is not a leaf class with only one immediate ancestor after the deleting.

   The removing class C_{del} is a leaf class. CA deletes the key K_{del} and public parameter PB_{del} of C_{del}. At last CA modifies the authorized relationship to remove C_{del}.

b. The immediate ancestor of the removing class is a leaf class with only one immediate ancestor after the removing.

The removing class C_{del} is a leaf class with only one immediate ancestor C_r. CA deletes the key K_{del} and public parameter PB_{del} of C_{del}. CA deletes the key K_r and the prime pair (e_r, d_r) and modifies the secret keys of C_r's ancestors. Then, CA generates random key and XOR public parameter for C_r as Step 5 in the key generation phase. At last CA modifies the authorized relationship to remove C_{del}.

2) Removing a Leaf Class with Multiple Immediate Ancestors in the Hierarchy:

The removing class C_{del} is a leaf class with multiple immediate ancestor $C_{r_1}, \cdots, C_{r_t}$. CA deletes the key K_{del} and the prime pair (e_{del}, d_{del}) of C_{del}. Then, CA does a test about the class C_{r_1}. If C_{r_1} is a leaf class with only one immediate ancestor after the removing, CA deletes the secret key and prime pair of C_{r_1}, and generates random key K_{r_1} and modifies public parameter PB_{r_1} for C_{r_1} as Step 5 in key generation phase. Then, CA modifies the secret keys of C_{del}'s ancestors. If C_{r_1} is not a leaf class with only one immediate ancestor after the removing, CA only modifies the secret keys of C_{del}'s ancestors. CA does the same test on the class $C_{r_2}, \cdots, C_{r_t}$. At last CA modifies the authorized relationship to remove C_{del}.

3) Removing a Non-leaf Class:

a. All immediate successors of the removing class are not leaf classes with only one immediate ancestor after the deleting.

The removing class C_{del} is a non-leaf class. CA deletes the key K_{del} and the prime pair (e_{del}, d_{del}) of C_{del}. Then, CA modifies the secret keys of C_{del}'s ancestors. At last CA modifies the authorized relationship to remove C_{del}.

b. Any immediate successor of the removing class is a leaf class with only one immediate ancestor.

The removing class C_{del} is a non-leaf class and $C_{r_1}, \cdots, C_{r_t}$ are the immediate successors of C_{del} with only one immediate ancestor after the removing. CA deletes the key K_{del} and the prime pair (e_{del}, d_{del}) of C_{del} and modifies the secret keys of C_{del}'s ancestors. Then, CA generates new random keys and XOR public parameters for $C_{r_1}, \cdots, C_{r_t}$ as Step 5 in key generation. At last CA modifies the authorized relationship to remove C_{del}.

4 The Efficiency Analysis Between Lo's and Our Scheme

The efficiency comparisons are comprised of the number of changed public parameters in dynamic key management, space complexity of storage for public parameters and time complexity of public key generation in key derivation.

4.1 The Number of Changed Public Parameters in Dynamic Key Management

In general scene, CA modifies the public parameters of the changed node and its all ancestors in Lo's scheme, but only the public parameters of the changed node and its immediate ancestors in our improved scheme. So, our scheme has less modification about public parameters in the dynamic management than Lo's scheme. Firstly, the computation of changed public parameters which are executed on CA are reduced. Then, it is easily inferred that the bottle of the interaction is alleviated between CA and trusted public platform.

4.2 The Efficiency Comparison on Space Complexity of Public Parameters Storage

In Lo's scheme, the public parameter is a product from a series of large prime numbers, where every one is coprime with $\varphi(m)$ and satisfies the condition $m^{7/8} \leq e_i \leq m$ [3]. Two disadvantages on storage are inevitable. The first one is that the storage of public parameter for each class must be provided as the largest product $PB_i = \prod_{allC_l, C_l \preceq C_i} e_l$. Obviously, the storage space is very large and more wasteful because many public parameters are shorter than the largest public parameters. The other one is the stability. When the hierarchical access control layer adds in dynamic key management, the length of largest public parameter adds. CA must modify the length of storage on public parameter for each class. With the operation, the storage of public parameters changes deeply. Not only more interaction between CA and trusted public platform but also lower storage stability on trusted public platform have to be considered.

In our improved scheme, the public parameters are divided into two parts as Table 3. The first part is about identity, which is the input of Hash function, so the length of output about Hash function is fixed. So, the storage on public parameter for non-leaf class and leaf class with two or more immediate ancestors are cancelled. The other part is XOR public parameter, which is related with the length of private key, but the private key is no more than m bits. So, the storage on public parameter for each class is m bits. Now, only one base table are considered for the classes with only one immediate ancestor. Because of the same length public parameters, the utilization of storage space is improved greatly. The public parameter is constant size, so the re-operation about the length of storage on public parameter for each node does not exist in dynamic key management. It is good for the system stability.

Table 3: The comparison on length of public parameter

Schemes	The public parameter on leaf class with only one immediate ancestor	The public parameter on non-leaf class or leaf class with more immediate ancestors
Lo's	m bits	$t_i \cdot m$ bits, where t_i denotes the number of C_i and C_i's successors except leaves with only one immediate ancestor
Our	m bits	0 bits

4.3 The Efficiency Comparison on Time Complexity of Key Derivation

For a simple description, a POSET is defined as following: C_j, C_i has t_j, t_i child nodes, respectively, and they have the relationship $C_i \preceq C_j$ and $t_i \leq t_j$. A division must be executed between C_j's and C_i's public parameters for the derivation key in Lo's scheme. It is a division of $t_j e$ bits, because of the relationship $C_i \preceq C_j$ and $PB_j = \prod_{allC_l, C_l \preceq C_j} e_l$. The latest research transforms one time division to the equal length multiplication in the time complexity $O(n^{\log_2 3})$ [18], so this computation is $t_j e$ bits multiplication in the time complexity $O(n^{\log_2 3})$. In our improved scheme, it is an e bits accumulative multiplication in $t_j - t_i$ times, because of the derivation key $z_{ji} = \prod_{allC_l, C_l \preceq C_j \wedge C_l \npreceq C_i} e_l$. So this computation is no more than $(t_j - t_i)e$ bits multiplication in the time complexity $O(n)$. Obviously, our improved scheme is more efficient than Lo's scheme because of $O(n) < O(n^{\log_2 3})$ and $(t_j - t_i)e \leq t_j e$ as Table 4.

Table 4: The comparison of time complexity

Schemes	The length of lager number	The time complexity of computation
Lo's	$t_j \cdot e$ bits	$O(n^{log_2 3})$
Our	$(t_j - t_i) \cdot e$ bits	$O(n)$

5 Proof of Security

Theorem 2. *The improved scheme is secure on the key recovery model in Definition 1.*

Proof. Challenger C constructs K'_{v^*}, but attacker A only have a negligible advantage to distinguish the K'_{v^*} and K_{v^*}.

Algorithm $STAT_v^{our}(1^\tau, G', ID, corr'_v)$:

1) Let us construct an input for $STAT_v^{our}(1^\tau, G', ID, corr'_v)$, where $g_{HL} = g_{our}$.
   - $G = G'$;
   - $pr = H(ID)$;
   - $u = v$ is the attacked class;
   - $corr_u = corr'_v$ are the keys of the classes which are corrupted.
2) Let the input of $STAT_v^{our}$ is $(1^\tau, G', ID, corr'_v)$ and the output is K_u^{our}.
3) The output of the classes except the leaf class with only one immediate ancestor are $K_u^{HL} = K_u^{our}$, which is corresponding to $K_{v*}^{HL'} = K_{v*}^{our'}$. Thus,
   $$Pr_{our}[k'_{v*} = k_{v*}] = Pr_{HL}[k'_{v*} = k_{v*}].$$
   So, we can conclude that our scheme is have the same security with Harn-Lin scheme.
4) Assume that we have the result of $K_{v*}^{HL'} = K_{v*}^{our'}$ as 3). Concluding the front result and the random secret key K_i of leaf class which attacker A owns, we modify the public parameter of leaf class to $PB_i = K_i \oplus H_2(K_{v*}^{our'}, ID_i, ID_u) = K_i \oplus H_2(K_{v*}^{HL'}, ID_i, ID_u)$. Now, it is a negligible advantage to distinguish the K'_{v*} and K_{v*} for the attacker who only has the leaf class key . Thus,
   $$Pr_{our}[K'_{v*} = K_{v*}] = |Pr_{HL}[K'_{v*} = K_{v*}] + \varepsilon_{H^{-1}}|.$$
   So, we can conclude that our scheme is have the same security with Harn-Lin scheme.

Combining the above result, we can conclude that our scheme has the same security with Harn-Lin scheme. The security proof of Harn-Lin scheme on the key recovery model is provided in [7]. So, our improved scheme is secure on the key recovery model. This concludes the Theorem 2. □

6 Conclusion

The paper proposes an improved ID-based hierarchical cryptography access control scheme which the public platform only store constant size of public parameter for the leaf classes with only one immediate ancestor. The improved scheme does a great optimization as following: The first one is the storage space of public parameters. The second one is the system stability on public platform part in key dynamic management, the third one is the information interaction between CA and the public platform and the last is the efficiency of key derivation. Then, to reduce the storage space further, the paper introduces ID as a part of public parameter. Comparing with the same type schemes, the improved ID-based scheme has high efficiency on space and time complexity and less interaction between CA and the public platform of trusted. At last, the paper does the work of security analysis for the improved ID-based scheme on the key recovery model.

7 Acknowledgments

This work is supported by Science and Technology on Communication Security Laboratory Foundation (NO.9140C110301110C1103) and The Weaponry Equipment Pre-Research Foundation, the PLA General Armament Department (NO.9140A04020311DZ02).

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems*, vol. 1, no. 3, pp. 239–248, 1983.

[2] G. Ateniese, A. D. Santis, L. A. Ferrara, et al. "Provably-secure time-bound hierarchical key assignment schemes," *Journal of Cryptology*, vol. 25, no. 2, pp. 243–270, 2012.

[3] J. Blömer, A. May, "New partial key exposure attacks on RSA," in *Advances in Cryptology (Crypto'03)*, pp. 27–43, Springer, 2003.

[4] M. Y. Chen, C. W. Liu, and M. S. Hwang, "Securedropbox: a file encryption system suitable for cloud storage services," in *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, pp. 21, 2013.

[5] M. Y. Chen, C. C. Yang, and M. S. Hwang, "Privacy protection data access control," *International Journal of Electronics and Information Engineering*, vol. 15, no. 6, pp. 411–419, 2013.

[6] H. Y. Chien, "Efficient time-bound hierarchical key assignment scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 16, pp. 1301–1304, 2004.

[7] P. D'Arco, A. De Santis, A. L. Ferrar, et al. "Variations on a theme by Akl and Taylor: Security and tradeoffs," *Theoretical Computer Science*, vol. 411, no. 1, pp. 213–227, 2010.

[8] S. M. El-Sayed, H. M. A. Kader, M. M. Hadhoud, and D. S. AbdElminaam, "Mobile cloud computing framework for elastic partitioned/modularized applications mobility," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 53–63, 2014.

[9] D. Giri and P. D. Srivastava, "A cryptographic key assignment scheme for access control in poset ordered hierarchies with enhanced security," *International Journal of Network Security*, vol. 7, no. 2, pp. 223–234, 2008.

[10] D. Giri and P. D. Srivastava, "An asymmetric cryptographic key assignment scheme for access control in tree structural hierarchies," *International Journal of Network Security*, vol. 4, no. 3, pp. 348–354, 2007.

[11] D. Giri and P. D. Srivastava, "An asymmetric cryptographic key assignment scheme for access control in tree structural hierarchies," *International Journal of Electronics and Information Engineering*, vol. 4, no. 3, pp. 348–354, 2007.

[12] L. Harn and H. Y. Lin, "A cryptographic key generation scheme for multilevel data security," *Computer & Security*, vol. 9, no. 6, pp. 539–546, 1990.

[13] M. S. Hwang, "Cryptanalysis of ycn key assignment scheme in a hierarchy," *Information Processing Letters*, vol. 3, no. 73, pp. 97–101, 2003.

[14] M. S. Hwang, J. W. Lo, and C. H. Liu, "Improvement on the flexible tree-based key management framework," *Computers & Security*, vol. 24, no. 6, pp. 500–504, 2005.

[15] M. S. Hwang and W. P. Yang, "Controlling access in large partially ordered hierarchies using cryptographic keys," *The Journal of Systems and Software*, vol. 67, no. 2, pp. 99–107, 1990.

[16] C. C. Lee, Y. M. Lai, and C. S. Hsiao, "Cryptanalysis of a simple key assignment for access control based on polynomial," *Journal of Information Security and Applications*, vol. 18, no. 4, pp. 215–218, 2013.

[17] I. C. Lin, M. S. Hwang, and C. C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy," *Future Generation Computer Systems*, vol. 4, no. 19, pp. 457–462, 2003.

[18] H. Liu and Z. Yu, *The time complexity of the large integer division depending on the large integer multiplication*, 2011. (`http://wenku.baidu.com/view/7d7f17120b4e767f5acfce32`)

[19] J. W. Lo, M. S. Hwang, and C. H. Liu, "A simple key assignment for access control based on polynomial," *The Arabian Journal for Science and Engineering*, vol. 38, no. 6, pp. 1397–1403, 2013.

[20] J. W. Lo, M. S. Hwang, and C. H. Liu, "An efficient key assignment scheme for access control in a large leaf class hierarchy," *Information Sciences*, vol. 181, no. 4, pp. 917–925, 2011.

[21] S. J. Mackinnon, P. D. Taylor, H. Meijer, and S. G. Akl, "An optimal algorithm for assigning cryptographic keys to control access in a hierarchy," *IEEE Transactions on Computers*, vol. 34, no. 9, pp. 797–802, 1985.

[22] M. Nikooghadam, A. Zakerolhosseini, and M. E. Moghadam, "Efficient utilization of elliptic curve crypto system for hierarchical access control," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1917–1929, 2010.

[23] V. Odelu, A. K. Das, and A. Goswami, "An effective and secure key-management scheme for hierarchical access control in e-medicine system," *Journal of Medical Systems*, vol. 37, no. 2, pp. 1–18, 2013.

[24] A. D. Santis, A. L. Ferrara, and B. Masucci, "Enforcing the security of a time-bound hierarchical key assignment scheme," *Information Sciences*, vol. 12, no. 176, pp. 1684–1694, 2006.

[25] T. H. Sun and M. S. Hwang, "A hierarchical data access and key management in cloud computing," *Innovative Computing, Information and Control Express Letters*, vol. 6, no. 2, pp. 569–574, 2012.

[26] S. F. Tzeng, C. C. Lee, and T. C. Lin, "A novel key management scheme for dynamic access control in a

hierarchy," *International Journal of Network Security*, vol. 12, no. 3, pp. 178–180, 2011.

[27] W. G. Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, no. 14, pp. 182–188, 2002.

[28] S. Y. Wang and C. S. Laihn, "Cryptanalysis of Hwang–Yang scheme for controlling access in large partially ordered hierarchies," *The Journal of Systems and Software*, vol. 75, no. 1, pp. 189–192, 2005.

[29] S. Wu and K. Chen, "An efficient key-management scheme for hierarchical access control in e-medicine system," *Journal of Medical Systems*, vol. 36, no. 4, pp. 1–13, 2012.

[30] J. Yeh, R. Chow, and R. Newman, "A key assignment for enforcing access control policy exceptions," in *Proceedings on International Symposium on Internet Technology*, pp. 54–59, 1998.

[31] J. Yeh, R. Chow, and R. Newman, "Key assignment for enforcing access control policy exceptions in distributed systems," *Information Sciences*, vol. 152, pp. 63–88, 2003.

[32] X. Yi and Y. Ye, "Security of tzeng's time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 15, pp. 1054–1055, 2003.

[33] Y. Zhou, J. Liu, H. Deng, et al. "Non-interactive revocable identity-based access control over e-healthcare records," in *Information Security Practice and Experience*, LNCS 9065, pp. 485–498, Springer, 2015.

Rang Zhou is a PH.D. student in the Department of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). He received his B.S. and M.S. degrees from the Department of Computer Science and Engineering, UESTC in 2009 and 2013, respectively, P.R.China. His research interests include: Access control security, cryptography and cloud security.

Chunxiang Xu is a professor in the Department of Computer Science and Engineering, UESTC. She received her B.S., M.S. and PH.D. degrees from XiDian University in 1985, 1988 and 2004, respectively, P.R.China. Her research interests include: Information security, cloud security and cryptography.

Wanpeng Li received his B.S. degree from Xihua University, P.R.China in 2010, and M.S. degree from UESTC P.R.China in 2013. He currently is a PH.D. student in the Department of Royal Holloway, University of London. His research interests include: Forward security and cryptography.

Jining Zhao received his B.S. degree from HeNan Normal University, P.R.China in 2009 and M.S. degree from UESTC P.R.China in 2013. He currently is a PH.D. student in the Department of Computer Science and Engineering, UESTC. His research interests include: Cloud auditing security and cryptography.

A Computational Review of Identity-based Signcryption Schemes

Murari Mandal[1], Gaurav Sharma[2], and Anil K. Verma[1]
(Corresponding author: Gaurav Sharma)
Computer Science and Engineering Department, Thapar University, Patiala-147004, India[1]
Amity School of Engineering, Amity University, Noida, India[2]
(Email: sharmagaurav86317@gmail.com)
(Received Jan. 20, 2015; revised and accepted July 4 & Dec. 7, 2015)

Abstract

Since 2002, several identity based signcryption schemes have been proposed. The purpose of designing a signcryption scheme is to perform signature and encryption both in one step but at lesser cost than performing signature and then encryption separately. In this paper, we present a literature survey on signcryption schemes for identity based setup. Our primary focus is on the schemes recently developed in standard model as the schemes in random oracle model are not actually practical. We present detailed comparison among the schemes based on computation cost, security features and suggest some final recommendation based on some future perspectives.

Keywords: Identity based cryptography, public key cryptography, signcryption, standard model

1 Introduction

The field of cryptography deals with providing various aspects of security for computer based communication [37]. For network based communications, confidentiality and authentication are the two most essential security features, which must be addressed. Confidentiality of message communicated between two or more users can be achieved through encryption. The properties of authentication (to confirm/verify the sender's identity), integrity (the message should not get altered before reaching the receiver) and non-repudiation (the sender can not deny the authorship of the message after the completion of the communication) are achieved by signatures.

The concept of identity based cryptography was first introduced by Shamir [33] in 1984. In ID based encryption/signature schemes, the identity of the user is used as the public key, or some well-known algorithms (or hash functions) are used to derive the same. Such an identity can be the email address, social security number or some string that can help to identify the user unambiguously. This alleviates the certificate management issue as the public key is implicitly authenticated. Although, the necessity of PKI (Public Key Infrastructure) is removed but this does require a PKG (Private Key Generator), which acts as a trusted authority to generate the private keys for the user with respect to their identity as and when requested by the user.

The implementation of ID Based Signature scheme (IBS) were presented by [13, 16] but, until 2001 the practical implementation of IBE (ID Based Encryption) was an open problem. Boneh and Franklin [7] presented the first practical IBE using bilinear pairing over elliptic curves.

Many other IBE schemes [6, 31, 34, 39] were proposed thereafter. In ID based encryption/signature schemes, some other security properties are also introduced such as public verifiability, forward secrecy etc. We will discuss about these terms in the later sections of this paper.

In 1997, Zheng [41] coined a term signcryption, which he derived by combining the words signature and encryption. The idea was to achieve signature and encryption both in a single logical step (in a single algorithm), which will cost less than the combined cost of performing signature and then encryption with the help of two separate algorithms. Zheng also presented a signcryption scheme based on Discrete Logarithm Problem (DLP).

Later, Baek et al. [9] presented the security proof for Zheng's scheme by introducing a security model. In 2002,

Malone-Lee [28] proposed the first identity based signcryption scheme including its security model. Later on, many signcryption schemes [4, 8, 11, 12, 27, 29] were presented. Most of these schemes were proven secure in the random oracle model by Bellare and Rogaway [5]. Although, the schemes provably secure in the random oracle model are quite efficient but the flaw in this model were pointed out in [2, 3, 10, 15]. Yu et al. [40] proposed the first identity based signcryption scheme without random oracle. Their scheme is based on Water's scheme [39]. Thereafter, several signcryption schemes were proposed in standard model. A survey of identity based signcryption was carried out by Li and Khan [22] by analysing ten signcryption schemes and their security parameters.

Since, the paper did not discuss much about the comparison among the signcryption schemes in standard model (only two schemes) as very few paper were published till that time. In this paper, we present a detailed analysis of the signcryption schemes and compare their efficiency and security properties.

The rest of the article is organized as follows. Section 2 contains preliminaries about the bilinear pairings. In Section 2.3, we give a general setup for signcryption scheme. In Section 3 we describe several security models. In Section 4 we give detailed analysis of the various signcryption schemes both in random oracle model and in standard model with help of the tables and in Section 5 we conclude our paper with some suggestions for future work.

2 Preliminaries

This section describes bilinear pairings and computational hardness problems, which are taken into consideration for the designing of an ID based signcryption schemes.

2.1 Bilinear Pairings

Let $\mathbb{G}_1$ be an additive group and $\mathbb{G}_T$ be a multiplicative group of prime order p. Then, bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, which satisfies the following conditions:

- *Bilinearity*: For all $X, Y, Z \in \mathbb{G}_1$, $\hat{e}(X + Y, Z) = \hat{e}(X, Z) \cdot \hat{e}(Y, Z)$ and $\hat{e}(X, Y + Z) = \hat{e}(X, Y) \cdot \hat{e}(X, Z)$;
- *Non-degeneracy*: $\hat{e}(X, X) \neq 1$;
- *Computability*: $\forall X, Y \in \mathbb{G}_1$, there is an efficient algorithm to compute $\hat{e}(X, Y)$.

Many pairing based cryptographic schemes use the bilinear pairing and depend on the intractability of some known problem like BDHP (Bilinear Diffie-hellman Problem), DBDHP (Decisional Bilinear Diffie-hellman Problem), CBDHP (Computational Bilinear Diffie-hellman Problem), etc.

2.2 Computational Hardness Problems

This section provides the computational hardness problems, which are used as a base to the security protocols.

Definition 1. Computational Diffie-Hellman Problem (CDHP)*: Given a generator P of $\mathbb{G}$ and aP, bP for unknown $a, b \in_R \mathbb{Z}_n^*$, the task of CDHP is to compute abP.*

Definition 2. Decisional Diffie-Hellman Problem (DDHP)*: Given a generator P of $\mathbb{G}$ and aP, bP, cP for unknown $a, b, c \in_R \mathbb{Z}_n^*$ the task of DDHP is to decide whether the equation $abP = cP$ holds.*

Definition 3. Gap Diffie-Hellman Problem (GDHP)*: Given a generator P of $\mathbb{G}$ and aP, bP for unknown $a, b \in_R \mathbb{Z}_n^*$ and an oracle DDHP (aP, bP, cP), which returns 1 if and only if $abP = cP$, the task of GDHP is to compute abP. The Gap Diffie-Hellman Assumption (GDHA) states that the probability of any polynomial-time algorithm solving the GDHP is negligible.*

2.3 General Setup

Lee [28] proposed the first IDSC (Identity Based Signcryption Scheme). We derive our general setup for IDSC from that scheme. The general setup for signcryption is explained in Table 1 and the process of secure exchange of message between the sender and receiver with the help of PKG is diagrammatically presented in Figure 1.

3 Security Models

Although, confidentiality and unforgeability are the two primary security requirements for any signcryption schemes but there are some special security properties like forward secrecy and public verifiability that have become essential for IDSC. In addition to this, Boyen [8] and Chow et al. [12] have defined few more security features like ciphertext unlinkability, ciphertext authentication, ciphertext anonymity and public ciphertext verifiability (see Figure 2). A single signcryption algorithm may not be able to ensure all these additional security features, as we will see that some of them contradict with each other. But, having some of these specialized security parameters might be very effective for security in a particular domain. We will discuss these security parameters in view of identity based signcryption setup. In the rest of this section we will give generalized definition of all the security parameters that we have mentioned above. Since every author defines his own security model based on which he gives his security results, our purpose is to give our readers a general idea about these security models.

3.1 Confidentiality

Confidentiality ensures that any third party (except the sender and receiver) will not be able to access or derive any information about the message being communicated. An et al. [1] suggested the notion of insider security and outsider security models. In the outsider security model, the adversary only has access to his own private key and he can signcrypt the message using the public keys of other users i.e. the adversary is having capability just like a user. Since, this is very weak assumption about an adversary so, we will neglect this case in the rest of our paper. In the insider security model, an adversary is given the power to perform adaptive chosen ciphertext attack also known as CCA2. It gives an access to adversary to unsigncrypt oracle, so that, he can unsigncrypt any ciphertext of his own choice (of course except the challenge ciphertext, otherwise life would be so easy for him, right!). In the rest of our paper we will only consider the insider security model.

Table 1: General setup for signcryption

Step	Action Performed
Setup	• For a given input 1^k, the PKG generates system parameters using some algorithm, where k is some security parameter • It also generates master public key mst_p and master private key mst_s. It keeps the master private key secret to himself
Extract	• For a given input (identity of the user), the PKG uses Extract algorithm to generate the private key and gives it to the user • The Extract algorithm will make use of the master private key mst_s for this purpose e.g. if user A with identity ID_A requests for a private key then, the private key $S_{ID_A} = Extract(ID_A)$
Signcrypt	• If ID_A wants to send a message m to ID_B, then the Signcrypt algorithm takes as input the message m, the private key of the sender ID_A and the identity of the receiver ID_B. The output ciphertext $\sigma = Signcrypt(m, S_{ID_A}, S_{ID_B})$
Unsigncrypt	• This algorithm takes the ciphertext σ as input, the identity of the sender ID_A and the private key of the receiver S_{ID_B} and returns a message m or symbol $\perp$ if the ciphertext is invalid one • Consistency check: If $\sigma = Signcrypt(m, S_{ID_A}, ID_B)$, then $m = Unsigncrypt(\sigma, S_{ID_B}, ID_A)$

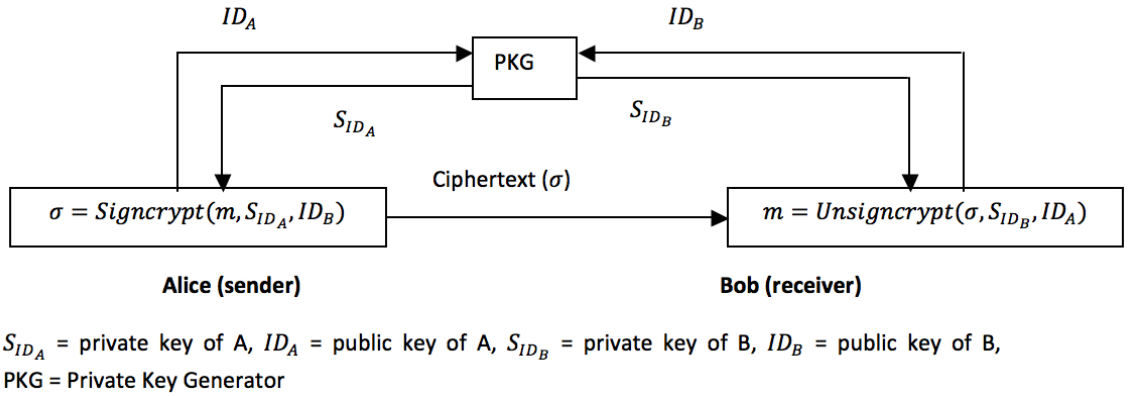


Figure 1: ID based signcryption

The following game is played between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

1) $\mathcal{C}$ runs the setup algorithm with security parameter k and gives the system parameters to the Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$.

2) **Phase 1:** $\mathcal{A}$ makes a polynomially bounded number of queries adaptively. By the term "Adaptively", we mean that every request can depend on the response to the previous query.

   a. *Key Extraction*: $\mathcal{A}$ gives an identity ID_A and $\mathcal{C}$ computes $S_{ID_A} = Extract(ID_A)$ and gives S_{ID_A} to $\mathcal{A}$.

   b. *Signcryption Queries*: $\mathcal{A}$ requests the challenger $\mathcal{C}$ to produce a signcryption on the message m by the sender ID_A to the receiver ID_B. Challenger responds with private key S_{ID_A} and $\sigma = Signcrypt(m, S_{ID_A}, ID_B)$.

   c. *Unsigncrypt Queries*: $\mathcal{A}$ requests to unsigncrypt a ciphertext σ with sender's identity ID_A and the receiver's identity ID_B to the challenger $\mathcal{C}$. The challenger responds with results as follows:

      i. The private key of ID_B. $S_{ID_B} = Extract(ID_B)$.

      ii. $UnSigncrypt(\sigma, S_{ID_B}, ID_A)$, the result can be the symbol $\perp$ in case of invalid ciphertext as input.

3) The adversary can make as many queries as he wants in the *Phase 1*, with restriction that he can't ask for the private key of the receiver of the actual ciphertext on which he is being challenged.

   a. The adversary chooses two plaintexts m_0, m_1 and the sender's identity ID_S and the receiver's identity ID_R on which he wants to be challenged. Remember that he can't make extraction query on ID_R in *Phase 1*.

   b. The challenger takes a random bit $b \in \{0,1\}$ and computes $\sigma^* = Signcrypt(m_b, S_{ID_S}, ID_R)$ and gives σ^* to $\mathcal{A}$.

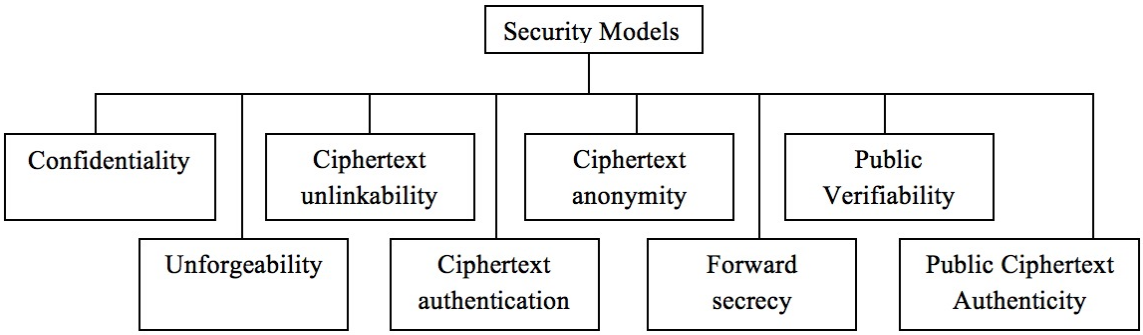


Figure 2: Security models

4) **Phase 2:** $\mathcal{A}$ can perform a polynomially bounded number of queries adaptively, similar to *Phase 1* but with restriction that he can't make key extraction query on ID_R or ID_S. Also, he can't make an unsigncrypt query on σ^*. $\mathcal{A}$ produces a bit b' and wins the game if $b' = b$. The advantage of $\mathcal{A}$ is defined as $Adv(\mathcal{A}) = |Probability(b' = b) - \frac{1}{2}|$.

Definition 4. *An ID based signcryption scheme IDSC is IND-CCA2 (Indistinguisability against adaptively Chosen Ciphertext Attack) secure if a PPT adversary $\mathcal{A}$ doesn't have non-negligible advantage in the above game.*

3.2 Unforgeability

This property ensures that the adversary can't produce the similar signature as of the challenger on a given message. For IDSC we consider the following game played between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. The Steps 1 to 4 of previous section will be again repeated for this game. So, we will directly discuss the 5th step. $\mathcal{A}$ generates a new triple (σ^*, ID'_S, ID'_R) i.e. a triple, which was not generated by the signcryption oracle. The adversary $\mathcal{A}$ was not allowed to make request for the private key of ID'_S during the *Phase 1* of the game. At the same time the adversary is allowed to request for the private key of the receiver, this will prevent a dishonest receiver to make forgery of the sender.

$\mathcal{A}$ wins the game if $UnSigncrypt(\sigma^*, S_{ID'_R}, ID'_S) \neq \perp$.

Definition 5. *An IDSC scheme is EU-CMA (Existential Unforgeability against Chosen Message Attack) secure if for all PPT adversaries, the advantage of $\mathcal{A}$ in the above game i.e. $Adv(\mathcal{A})$ = Probability of success in above game is negligible.*

3.3 Ciphertext Unlinkability

This property gives the sender of a message the power to deny having sent a message to the receiver even if that message might contain the signature of the sender only. It means that although the message is signed by the sender only but whether the ciphertext was sent by the sender or not can't be verified by anyone. Such property may be very helpful in certain situations such as security agents communicating with their base station from and some other cases as mentioned by Boyen [8].

3.4 Ciphertext Authentication

This property is kind of an exception case of Ciphertext Unlinkability. The receiver can authenticate that the ciphertext indeed came from the sender who has signed the message that it contains but he can't prove it to anyone. The detailed description is given by Boyen [8].

3.5 Ciphertext Anonymity

This property makes the ciphertext anonymous i.e. except the receiver no one should be able to know about the author or recipients of the message.

3.6 Forward Secrecy

Forward secrecy means that even if a private key of the sender gets compromised, still it will not be possible for someone to unsigncrypt the messages that were signcrypted previously by the user [12]. In the insider security model, forward secrecy is naturally achieved because if a scheme is CCA2 secure then it will also provide forward secrecy.

3.7 Public Verifiability

This property states that if a ciphertext is provided by the receiver and also the corresponding message and some other information to a trusted third party, the third party should be able to verify that the ciphertext is valid signature on the message by the sender even if the private key of the sender is not available.

3.8 Public Ciphertext Authenticity

This notion was presented by Chow et al. [12]. This property makes it possible for any third party to verify the ciphertext's origin and also to check its validity. The third

party is not allowed to get any information from the receiver. This property directly contradicts with the ciphertext authenticity definition above, so it is not possible to achieve both of them simultaneously in one scheme.

4 Analysis of Identity Based Signcryption Schemes

We divide our discussion in two parts. First we will discuss about the schemes that are proposed under random oracle model and then about the schemes proposed in standard model.

4.1 Identity Based Signcryption Schemes in Random Oracle Model

After the practical implementation of ID based encryption by Boneh and Franklin [7], Malone-Lee [28] proposed the first ID based signcryption scheme. He also presented a security model to prove its security. But, Libert and Quisquater [27] pointed out that since the signature on the plaintext is visible in the ciphertext, therefore the scheme can not ensure confidentiality of the message. They proposed three new schemes, the first one alleviated the semantic security issue, in the second scheme they modified the previous scheme to produce shorter ciphertext and in the last scheme they added the forward secrecy property but by doing this the scheme lost the public verifiability property. They further proposed an open problem to construct a signcryption scheme that provides both forward secrecy and public verifiability. In 2004, this problem was solved by Chow et al. [12]. They not only designed a new scheme that provides both forward secrecy and public verifiability but also added a new security property called public ciphertext authenticity. In the same year McCullagh and Barreto [29] also presented a new scheme to address the same issue.

In 2003, Boyen [8] introduced some specialized security parameters, such as, ciphertext unlinkability, ciphertext authentication, ciphertext anonymity and presented a scheme to achieve a two layer sign and then encrypt combination. This scheme facilitates multi-recipient signcryption i.e. encrypting the same message with a shared signature and also single bulk message encryption. In 2005, Chen and Malone-Lee [11] improved Boyen's [8] scheme and made it more efficient. Barreto et al. [4] improved it further to achieve the most efficient ID based signcryption scheme. In 2007, Li et al. [21] presented an efficient signcryption scheme with the property of ciphertext anonymity.

Since, security proof in random oracle models are not applicable in real time situations [2, 3, 10, 15], therefore it is important to design schemes that are secure in standard model. In the next section we discuss about such schemes.

4.2 Identity Based Signcryption Schemes in Standard Model

In 2009, Yu et al. [40] proposed the first ID based signcryption scheme in the standard model. They combined the ideas from Waters [39] and Paterson and Schuldt's [32] to design their new scheme. This scheme was proved insecure by Bo Zhang [42], Zhengping et al. [17], Wang and Qian [38] and Zhang et al. [43]. This scheme is vulnerable to IND-CCA2 attack and the SUF-CMA attack and therefore this scheme is neither semantically secure nor unforgeable. The improved scheme proposed by Zhengping et al. [17] was also cryptanalysed by Li et al. [14]. In 2010, Zhang [42] proposed a scheme, which was later proved IND-CCA2 insecure by Li and Takagi [24], thus attacking the semantic security of this scheme but it still provides unforgeability. Li and Takagi [24] improved Zhang's [42] scheme and proposed a new scheme, which was proven both IND-CCA2 and EUF-CMA insecure by Selvi et al. [35]. Further improvement given by Li et al. [25] was also proven insecure by Selvi et al. [35]. In 2011, another scheme was proposed by Li et al. [23] in which they achieved both confidentiality and unforgeability at less computational cost in comparison to previous schemes. But flaws in their proof for security against IND-CCA2 attack were pointed out by Selvi et al. [35]. Selvi et al. [35] presented a signcryption scheme by direct combination of IBE and IBS. They took the IBS in standard model proposed by Paterson and Schuldt [32] and IBE in standard model proposed by Kiltz and Vahlis [18]. They followed sign and then encrypt method as this is the only combination that is both IND-CCA2 and EUF-CMA secure. Although this scheme is secure but it does suffer from inefficiency.

In 2012, Li et al. [26] presented a fully secure ID based signcryption scheme, which is having shorter ciphertext than the previous schemes. They also compared the efficiency of their scheme with previous schemes and presented an analysis based on that. Such a scheme may be preferable in real time applications. But later, Ming and Wang [30] proved that their scheme is not semantically secure against chosen-message attacks and it is also not existentially unforgeable against chosen-message attacks. Lee et al. [20] presented a signcryption scheme that produced even shorter ciphertext in size compared to Li et al.'s scheme. Kushwah and Lal [19] present two ID based signcryption schemes, the first one provides the semantic security and unforgeability and the second scheme provides public ciphertext authentication. In 2012, Selvi et al. [36] proposed the most secure ID based signcryption scheme in standard model. Their security model fulfills the strongest notion for security in identity based signcryption schemes. Their scheme provides public ciphertext authenticity. But this increased the computational cost for the scheme. So, we can see that there is trade-off between the tightness of security and the efficiency of any signcryption scheme.

The computational costs of all the signcryption

schemes that have been discussed are tabulated in Table 2. In Table 2, in the pairing (Pair) column we have considered the total number of pairings required to either signcrypt or unsigncrypt. Multiplications (Mul), Exponentiations (Exp), Inverse (Inv), Addition/Subtraction (Add/Sub) are all performed in group (either in $\mathbb{G}_1$ or in $\mathbb{G}_T$). All these constitute to the cost of computation of a signcryption scheme. In the Hash column, only the number of hashing performed is listed and the type of hash function depends on the choice of the designer. ID_L denotes the bit length of all the identities and M_L denotes the bit length of the message. In a row, for example in the first row, the upper row describes the computational cost for signcryption and the lower row describes the cost in unsigncryption. ROM refers to "Random Oracle Model" and SM to "Standard Model".

The security analysis is presented in Table 3. The Cryptanalysis (C.A.) column describes by which author the cryptanalysis was done and the Attack (Att.) column describes which type of attack was made. The security parameters considered are Confidentiality (Con), Unforgeability (Unf), Public Verifiability (PuV), Forward Secrecy (FoS), Ciphertext Unlinkability (CiU), Ciphertext Anonymity (CiA), Ciphertext Authenticity (CiAu), Public Ciphertext Authenticity (PuCA).

5 Conclusion

Identity based signcryption has become a very important area of research as it performs both encryption and signature in one logic step and at lesser cost than direct combination of signature and encryption. By this survey we draw following conclusions:

1) Since random oracle models are not feasible to implement in real time applications, so schemes in standard model with tighter security and more efficiency, need to be designed.

2) The cost efficiency of signcryption can be very useful in areas such as wireless sensor networks, mobile ad hoc networks. Further new areas of implementation of ID based signcryption need to be explored.

3) The security of the latest IDSC schemes also has to be analysed. Since, most of the previous schemes in standard model has been cryptanalysed so it is important to thoroughly analyze the latest schemes before implementing them for practical purpose.

References

[1] J. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," in *Advances in Cryptology (Eurocrypt'02)*, LNCS 2332, pp. 83–107, Springer, 2002.

[2] B. Barak, "How to go beyond the black-box simulation barrier," in *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*, pp. 106–115, 2001.

[3] B. Barak, Y. Lindell, and S. Vadhan, "Lower bounds for non-black-box zero knowledge," in *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 384–393, 2003.

[4] P. Barreto, B. Libert, N. McCullagh, and J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Advances in Cryptology (Asiacrypt'05)*, LNCS 3788, pp. 515–532, Springer, 2005.

[5] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.

[6] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 417–42, 2008.

[7] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, pp. 586–615, 2003.

[8] X. Boyen. "Multipurpose identity-based signcryption," in *Advances in Cryptology (Crypto'03)*, LNCS 2729, pp. 383–399, Springer, 2003.

[9] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption," in *Public Key Cryptography*, LNCS 2274, pp. 80–98, Springer, 2002.

[10] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, vol. 51, pp. 557–594, July 2004.

[11] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Public Key Cryptography (Pkc'05)*, LNCS 3386, pp. 362–379, Springer, 2005.

[12] S. Chow, S. Yiu, L. Hui, and K. Chow, "Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity," in *International Conference on Information Security and Cryptology (Icisc'03)*, LNCS 2971, pp. 352–369, Springer, 2004.

[13] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology (Crypto'86)*, LNCS 263, pp. 186–194, Springer, 1987.

[14] L. Fagen, Q. Zhiguang, "Analysis of an identity-based signcryption scheme in the standard model," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94, no. 1, pp. 268–269, 2011.

[15] S. Goldwasser and Y. Kalai, "On the (in)security of the fiat-shamir paradigm," in *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 102–113, 2003.

[16] L. Guillou and J. Quisquater, "A "paradoxical" indentity-based signature scheme resulting

Table 2: Computational cost comparison

IDSC	Pair	Exp	Mul	Hash	Inv	Add/Sub	Model
Malone-Lee [28]	1	0	3	3	0	1	ROM
	4	1	0	2	0	0	
Libert and Quisquater* [27]	2	2	2	3	0	1	ROM
	4	2	2	3	0	0	
Libert and Quisquater* [27]	2	2	2	4	0	1	ROM
	4	2	2	4	1	0	
Libert and Quisquater* [27]	1	1	3	3	0	1	ROM
	2	0	2	3	0	0	
Boyen [8]	1	4	3	6	0	0	ROM
	4	2	1	7	0	0	
Chow et al.* [12]	2	0	2	2	0	0	ROM
	4	0	3	2	0	0	
Chen and Malone-Lee [11]	1	0	3	4	0	0	ROM
	3	0	1	4	0	0	
Barreto et al. [4]	0	1	3	3	0	1	ROM
	2	1	2	3	1	1	
McCullagh et al. [29]	0	1	2	2	0	0	ROM
	2	1	1	2	0	0	
McCullagh et al. [29]	0	1	3	2	0	0	ROM
	2	0	1	2	0	1	
Li et al. [21]	0	0	3	2	0	0	ROM
	4	0	1	2	0	0	
Yu et al.[40]	1	4	$ID_L + M_L + 2$	1	0	0	SM
	6	0	$ID_L + M_L + 4$	1	1	0	
Zhengping et al. [17]	1	4	$ID_L + M_L + 2$	1	0	0	SM
	6	0	$ID_L + M_L + 4$	1	1	0	
Zhang [42]	1	6	$ID_L + M_L + 3$	2	0	0	SM
	6	2	$ID_L + M_L + 5$	2	2	0	
Li and Takagi [24]	1	6	$2ID_L + M_L + 3$	2	0	0	SM
	6	2	$ID_L + M_L + 5$	2	1	0	
Li et al.* [25]	0	6	$ID_L + M_L + 1$	1	0	0	SM
	5	0	$ID_L + M_L + 3$	0	1	0	
Li et al. [26]	1	4	$ID_L + M_L + 2$	1	0	0	SM
	6	0	$ID_L + M_L + 4$	1	1	0	
Selvi et al. [36]	1	5	$ID_L + M_L + 3$	4	0	0	SM
	6	2	$ID_L + M_L + 5$	4	1	0	
Selvi et al.* [35]	1	8	$ID_L + M_L + 3$	2	0	0	SM
	6	3	$ID_L + M_L + 5$	1	0	0	
Li et al.* [23]	0	5	$ID_L + M_L + 3$	2	0	0	SM
	5	2	$ID_L + M_L + 6$	2	1	0	
Lee et al. [20]	0	7	$ID_L + M_L + 3$	4	0	0	SM
	4	2	$ID_L + M_L + 5$	3	1	0	
Kushwah and Lal [19]	1	4	$2ID_L + M_L + 1$	3	0	0	SM
	6	0	$ID_L + M_L + 3$	2	2	0	
Kushwah and Lal [19]	1	4	$ID_L + M_L + 2$	2	0	0	SM
	6	0	$ID_L + M_L + 4$	2	1	0	

* This scheme also uses a symmetric cipher.

Table 3: Security analysis

IDSC	Con	Unf	PuV	Fos	CiU	CiAn	CiAu	PuCA	C.A.	Att.
Malone-Lee [28]	N	Y	Y	Y	N	N	N	N	[27]!	1
Libert and Quisquater* [27]	Y	Y	Y	N	N	N	N	N		
Libert and Quisquater* [27]	Y	Y	Y	N	N	N	N	N		
Libert and Quisquater* [27]	Y	Y	N	Y	N	N	N	N		
Boyen [8]	Y	Y	Y	Y	Y	Y	Y	N		
Chow et al.* [12]	Y	Y	Y	Y	N	N	N	Y		
Chen and Malone-Lee [11]	Y	Y	Y	Y	Y	Y	Y	N		
Barreto et al. [4]	Y	Y	Y	Y	N	N	N	N		
McCullagh et al. [29]	Y	Y	N	Y	N	N	N	N		
McCullagh et al. [29]	Y	Y	Y	Y	N	N	N	N		
Li et al. [21]	Y	Y	Y	Y	N	Y	N	N		
Yu et al.[40]	N	N	Y	Y	N	N	N	N	[38, 42, 43, 17]	3, 4
Zhengping et al. [17]	N	N	Y	Y	N	N	N	N	[14]	1, 2
Zhang [42]	N	Y	Y	Y	N	N	N	N	[24]	1
Li and Takagi [24]	N	N	Y	Y	N	N	N	N	[35]	1, 2
Li et al.* [25]	N	N	Y	Y	N	N	N	N	[35]	1, 2
Li et al. [26]	Y	Y	Y	Y	N	N	N	N	[30]	1, 2
Selvi et al. [36]	Y	Y	Y	Y	N	N	Y	N		
Selvi et al.* [35]	Y	Y	Y	Y	N	N	N	Y		
Li et al.* [23]	Y#	Y	Y	Y	N	N	N	N	[35]	
Lee et al. [20]	Y	Y	Y	Y	N	N	N	N		
Kushwah and Lal [19]	Y	Y	N	Y	N	N	N	N		
Kushwah and Lal [19]	Y	Y	Y	Y	N	N	N	Y		

! Security weakness in semantic security was pointed out. # Flaw in the security proof as pointed out by [35]. 1→IND-CCA2, 2→EUF-CMA, 3→IND-IDSC-CCA [40], 4→EUF-IDSC-CMA [40].

from zero-knowledge," in *Advances in Cryptology (Crypto'88)*, LNCS 403, pp. 216–231, Springer, 1990.

[17] Z. Jin, Q. Wen, H. Du, "An improved semantically-secure identity-based signcryption scheme in the standard model," *Computers & Electrical Engineering*, vol. 36, no. 3, pp. 545–552, 2010.

[18] E. Kiltz and Y. Vahlis, "CCA2 secure IBE: Standard model efficiency through authenticated symmetric encryption," in *Topics in Cryptology (Ctrsa'08)*, LNCS 4964, pp. 221–238, Springer, 2008.

[19] P. Kushwah and S. Lal, "Provable secure identity based signcryption scheme without random oracles," *International Journal of Network Security & Its Applications*, vol. 4, no. 3, pp. 97 – 110, 2012.

[20] P. Lee, P. Udaya, and N. Shivaramakrishnan, "Efficient identity-based signcryption without random oracles," in *Proceedings of the Tenth Australasian Information Security Conference*, vol. 125, 2012.

[21] C. Li, G. Yang, D. Wong, X. Deng, and S. Chow, "An efficient signcryption scheme with key privacy," in *Public Key Infrastructure*, LNCS 4582 of, pp. 78–93, Springer, 2007.

[22] F. Li and M. Khan, "A survey of identity-based signcryption," *IETE Technical Review*, vol. 28, no. 3, pp. 265–272, 2011.

[23] F. Li, F. Muhaya, M. Zhang, and T. Takagi, "Efficient identity-based signcryption in the standard model," in *Provable Security*, LNCS 6980, pp. 120–137, Springer, 2011.

[24] F. Li and T. Takagi, "Secure identity-based signcryption in the standard model," *Mathematical and Computer Modelling*, vol. 57, no. 11C12, pp. 2685–2694, 2013.

[25] F. Li, L. Yongjian, Q. Zhiguang, and T. Takagi, "Further improvement of an identity-based signcryption scheme in the standard model," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 413–421, 2012.

[26] X. Li, H. Qian, J. Weng, and Y. Yu, "Fully secure identity-based signcryption scheme with shorter signcryptext in the standard model," *Mathematical and Computer Modelling*, vol. 57, no. 3C4, pp. 503 – 511, 2013.

[27] B. Libert and J. Quisquater, "New identity based signcryption schemes from pairings," IACR Cryptology ePrint Archive, Report 2003/023, 2003.

[28] J. Malone-Lee, "Identity-based signcryption," IACR Cryptology ePrint Archive, Report 2002/098, 2002.

[29] N. McCullagh and P. Barreto, "Efficient and forward-secure identity-based signcryption," IACR Cryptology ePrint Archive, Report 2004/117, 2004.

[30] Y. Ming and Y. Wang, "Cryptanalysis of an identity based signcryption scheme in the standard model," *International Journal of Network Security & Its Applications*, vol. 18, no. 1, pp. 165–171, 2016.

[31] D. Naccache, "Secure and practical identity-based encryption," *IET Information Security*, vol. 1, no. 2, pp. 59–64, 2007.

[32] K. Paterson and J. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Information Security and Privacy*, LNCS 4058, pp. 207–222, Springer, 2006.

[33] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, LNCS 196, pp. 47–53, Springer,1985.

[34] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (Eurocrypt'05)*, LNCS 3494, pp. 457–473, Springer, 2005.

[35] S. Selvi, S. Vivek, D. Vinayagamurthy, and C. Rangan, "On the security of id based signcryption schemes," IACR Cryptology ePrint Archive, Report 2011/664, 2011.

[36] S. Selvi, S. Vivek, D. Vinayagamurthy, and P. Rangan, "ID based signcryption scheme in standard model," in *Provable Security*, Lecture Notes in Computer Science, pp. 35–52, Springer Berlin Heidelberg, 2012.

[37] G. Sharma, S. Bala, and A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," vol. 18, no. 1, pp. 82–89, 2016.

[38] X. Wang and H. Qian, "Attacks against two identity-based signcryption schemes," in *IEEE Second International Conference on Networks Security Wireless Communications and Trusted Computing*, vol. 1, pp. 24–27, 2010.

[39] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology (Eurocrypt'05)*, LNCS 3494, pp. 114–127, Springer, 2005.

[40] Y. Yu, B. Yang, Y. Sun, and S. Zhu, "Identity based signcryption scheme without random oracles," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 56–62, 2009.

[41] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)," in *Advances in Cryptology (Crypto'97)*, LNCS 1294, pp. 165–179, Springer, 1997.

[42] B. Zhang, "Cryptanalysis of an identity based signcryption scheme without random oracles," *Journal of Computational Information Systems*, vol. 6, no. 6, pp. 1923–1931, 2010.

[43] M. Zhang, P. Li, B. Yang, H. Wang, and T. Takagi, "Towards confidentiality of ID-based signcryption schemes under without random oracle model," in *Intelligence and Security Informatics*, LNCS 6122, pp. 98–104, Springer, 2010.

Anil Kumar Verma is currently a faculty in the department of Computer Science and Engineering at Thapar University, Patiala. He received his B.S., M.S. and Doctorate in 1991, 2001 and 2008, respectively, majoring in Computer Science and Engineering. He has worked as Lecturer at M.M.M. Engg. College, Gorakhpur from 1991 to 1996. He joined Thapar University in 1996 and is presently associated with the same Institute. He has been a visiting faculty to many institutions. He has published over 100 papers in referred journals and conferences (India and Abroad). He has chaired various sessions in the International and National Conferences. He is active member of MIEEE, MACM, MISCI, LMCSI, MIETE, GMAIMA. He is a certified software quality auditor by MoCIT, Govt. of India. His research interests include wireless networks, routing algorithms and mobile clouds.

Gaurav Sharma received his PhD and M.E degree in Computer Science & Engineering from Thapar University, Patiala, India. He had received M. Sc. as well as B. Sc. degree from CCS University, Meerut, India. He is an active member of IEEE and ACM. Presently he is working as an Asst. Professor at Amity University, India. His area of interests is routing and security in Ad hoc networks.

Murari Mandal was born in India, in 1990. He received the B.E. degree in Computer Science from BITS Pilani, India, in 2011 and M.E. degree in Software Engineering from Thapar University, India, in 2015. He has worked as Research Assistant in Computer Science department of BITS Pilani from 2015 to 2016. In 2016, he joined the Computer Engineering Department, MNIT Jaipur, as a Research Scholar. His current research interests include computer vision, image/video processing, machine learning, real-time systems, cryptography and sensor networks.

Analysis of Algorithms for Overlapping Resource Access Members in Cloud Computing

Amar Buchade[1], Rajesh Ingle[2]
Department of Computer Engineering, College of Engineering, Pune[1]
Wellesely Rd, Shivajinagar, Pune, Maharashtra 411005, India
Department of Computer Engineering & Pune Institute of Computer Technology[2]
Sr. No 27, Pune-Satara Road, Dhankawadi, Pune, Maharashtra 411043, India
(Email: arb.comp@coep.ac.in, ingle@ieee.org)
(Received Aug. 12, 2015; revised and accepted Nov. 12 & Dec. 15, 2015)

Abstract

In Cloud computing environment, resources include virtual machine, CPU and Storage. These resources are accessed by tenants. Group key may be used to access the resources securely. Group key is constructed using tree by considering tenants in a group. In existing scenario, different key trees are formed even if tenants are common among multiple groups to access the resources. This paper addresses the issues of overlapping tenants that accesses resources. If there are overlapping members in multiple groups, combined key trees may be formed. Through the analysis, it is observed that computational overhead is decreased by 24% if we combine the key trees than the separate key trees. It is also observed that key establishment time for combined key trees is less compared to separate key trees.

Keywords: Computational cost, key tree, resource, resource access membership matrix

1 Introduction

In cloud computing environment, resources are considered as virtual machine, CPU, storage. These resources are accessed by multiple tenants. Users of facebook may share data (multiple files) in multiple groups. Members in a group accesses the resources. To protect the resource from unauthorized users, each member in the group shares the partial information for forming the group key. In present scenario, group key is formed by considering separate key trees even if members are common to access multiple resources. It incurs redundant operations and thus leads to increase in computational cost and key establishment time. It causes delay in accessing the actual resource which an obviously violates the feature of cloud computing such as on demand resource access. Thus our paper proposes combined key trees formation and its analysis for the tenants overlapped in multiple resources.

Other example, member can be a part of multiple projects. Multiple tenants can be involved in multiple projects. For security purpose, members in a group form the group key to access the resource.

The other examples can be users of whatsapp/facebook sharing multiple files in groups. Many members can be overlapped in groups to access the files.

The solution is to combine key trees for resources which containing common members. We prove that our approach is efficient than the forming separate key trees for overlapping resources access members.

It reduces computational overhead and group key establishment time. It helps to support on demand resource access property of the cloud computing.

To form the group key, TGDH protocol is used [9, 12, 13]. More specifically our contributions are

1) Illustration of the algorithms through the examples.
2) Computation cost analysis of resource key formation for separate key trees and combined key tree in terms of total number of sequential exponentiation operations.
3) Formulation of key establishment time and analysis.

The paper is organized as follows. In Section 2, we describe about resource. Section 3 describes combining key trees algorithm in brief, Section 4 presents computational cost details in terms of sequential exponential and key establishment time, Section 5 covers results and analysis and Section 6 presents conclusion.

Assumption: Tree based Group Diffie Hellman Protocol [13] is used. Member who is acting as sponsor assumed to be trustworthy. The words "tenant" and "member" used alternatively.

2 Resource

2.1 Initializations

Let Resource group R = $\{R_1, R_2, R_3, R_4, ...R_n\}$;

Consider two resources R1 and R2.

Let $\{m_1, m_2, m_3, m_4, ...m_n\}$ be the members accessing resource R1.

Let $\{n_1, n_2, n_3, n_4, ...n_n\}$ be the members accessing resource R2.

It is possible to have members overlapped to access the resources R1 and R2.
Assume $R1 \cap R2 =$ cm where cm is number of overlapping members which accesses the resources R1 and R2.

2.2 Resource Key Tree

Figure 1 shows resource key tree with leaf nodes represents group members m1, m2, m3, m4 etc. [8].

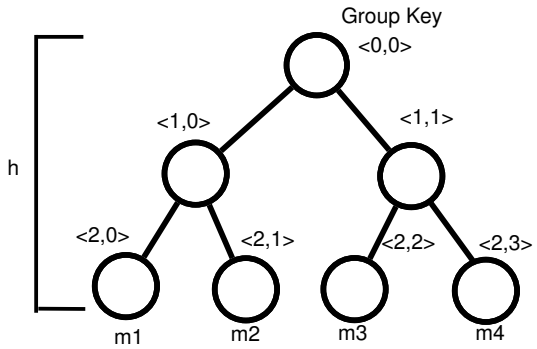


Figure 1: Resource key tree

Diffie and Hellman presented two party key exchange protocol called TGDH in 1976 [13].

- In TGDH [1, 6, 22, 23] group key is formed from bottom-up fashion.
- Members m1, m2, m3 and m4 have $\alpha 1, \alpha 2, \alpha 3$ and $\alpha 4$ private keys, respectively.
- Each member forms the public key called as blinded key.
- In this case, g is generator, p is prime number Member blinded keys are $g^{\alpha 1} \bmod p$, $g^{\alpha 2} \bmod p$, $g^{\alpha 3} \bmod p$, and $g^{\alpha 4} \bmod p$.
- Each member knows all keys on key path and all blinded keys. Key path of m2 includes the node at $< 1, 0 >$ and node at $< 0, 0 >$.
- Thus resource group key is formed as below.

$$g^{\alpha 1.\alpha 2.\alpha 3.\alpha 4} \bmod p$$

Let h is the height of the key tree. From Figure 1, h=2.

Number of exponential operations performed serially by the member are called sequential exponentiation operations. It is observed that number of sequential exponentiation required to form the group key = 2h - 2.

Thus to calculate group key at node $< 0, 0 >$, member m2 two sequential exponential operations mainly at node $< 1, 0 >$ and node $< 0, 0 >$ are required.

2.3 Resource Access Membership Matrix

Every tenant that is part of the resource, entry is made in its resource access membership [RAM] matrix also for any member that joins/leaves the resource. RAM matrix contains the following entries.

Members m_1, m_2, m_3, ..., m_n represents the entries in row wise.

Resources R_1, R_2, R_3, ..., R_n represents the entries in column wise.

$$\begin{bmatrix} 1 & 1 & \cdots & \cdots \\ 1 & 0 & \cdots & \cdots \\ 1 & 1 & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \end{bmatrix}$$

It shows that there are $R_1, R_2, \ldots, R_n$ resources. Member m_1 accesses resource R_1 and R_2 i.e. overlapped to access the resources R_1 and R_2 while member m_2 is part of resource R_1, m_3 is a part of resources R_1 and R_2. Three dots (...) indicates entry either 0 or 1.

3 Combining Resource Key Trees Algorithm

In existing key management algorithm [11, 16, 20, 24], separate key tree is built for each resource, even if same members are accessing multiple resources. Buchade and Ingle gives combining key tree algorithm [4]. This algorithm takes consideration of resource access membership matrix. Key tree of overlapping members are formed as well as key tree of non overlapping members are formed. These key trees are combined. The combined key tree of overlapping members is rooted at the root node to reduce the height of the tree.

4 Computational Cost

This section covers the proof of sequential exponentiation and key establishment time for combined and separate key trees.

Lemma 1. *Total number of sequential exponentiation (SE) for separate key trees (SKT) required more than combined key tree (CKT).*

Proof. Total number of sequential exponentiation operations with SKT

$$[SESKT] = \sum_{i=1}^{NRK} (2h_i - 2)N_i \tag{1}$$

where $h_i = log(N_i)$ (property of binary tree); NRK = Total number of resource key tree; h_i = height of i^th resource key tree; N_i = Number of members of i^th resource key tree. Average height of each member, $h = \log(N)$.

Total number of SE due to overlapping members:

$$[SEOM] = \sum_{j=1}^{NOT} (2h_j - 2)O_j, \quad (2)$$

where $h_j = log(O_j)$, $h > 1$ otherwise SE=1; NOT = Number of trees formed due to overlapping members; h_j = height of j^th resource key tree; O = Number of overlapping members; Average height of each member, $h = \log(O)$.

SE due to combined key trees [SECKT]
= SE due to SKT - SE due to Overlapping members.

Thus it is observed that SE for separate key trees is more compared to combined key trees.

□

Lemma 2. *Key establishment time for SKTs is more than key establishment time for CKT. It depends on Number of members overlapped to access the resources.*

Proof. For SKT, total Number of members,

$$N = \sum_{i=1}^{NRK} N_i$$

Average height of each member, $h = \log(N/NRK)$; Time required to form the group key =

$$(2h - 2) * DH_t = (2log(N/NRK) - 2) * DH_t, \quad (3)$$

where, NRK = Number of resource key trees; N_i = Number of members of i^{th} resource key tree; DH_t = Time required to perform one Diffie Hellman Exponentiation operation.

For CKT, total Number of members = $N - cm$, where cm = Number of overlapping members; Average height of each member, $h = \log((N - cm)/NRK)$; Time required to form the group key =

$$(2h - 2) * DH_t = (2log((N - cm)/NRK) - 2) * DH_t \quad (4)$$

From Equations 3 and 4, it is observed that key establishment time for SKTs can be more than key establishment time for CKT. It depends on number of overlapping members. □

4.1 Single Join

Buchade and Ingle stated algorithm when tenant wants to access the resource [4]. If the joining tenant is not overlapped, it is added in the key tree as per TGDH. If it is overlapped to other resources forms the key graph. The details of the algorithm is given in detail through the examples below.

4.1.1 Example: Single Member Join

This example illustrates how member m3 join to access the resource and how resource access membership matrix is maintained. Member uses TGDH [13] to join to access the resource.

1) There are two resources namely R1 and R2;
2) Member m1, m2 accesses the resources R1;
3) Member m5,m6 accesses the resources R2;
4) Each member has to maintain resource access membership matrix.

Example:

1) Member m3 joins R1.
2) Member m3 wants to access R2, broadcast join request for R2 alongwith message containing it is already having membership with R1.
3) Each member in R1 and R2 notices and makes the entry '1' against the entry of m3 in resource access membership matrix.
4) Sponsor of R2 gives/broadcast blinded keys, membership details.
5) Thus m3 is made sponsor. Because it is a member of R1 and R2.
6) Member m3 joins for resource R2 and builds key graph.
7) Member m3 builds key graph as it is overlapped with R1 and R2. It also makes the entry in resource membership matrix.
8) Each Member of R1 and R2 has its own tree view.
9) Each member of R1 except m3 has the following view of Resource Access Membership Matrix. R1=$m1, m2, m3$; Rows represents members m_1, m_2, m_3; and Columns represents Resources R_1, R_2.

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

It indicates that m1 represents that m1 is part of (access) of R1, m2 is part of R2 and m3 is part of R1 and R3.

10) Member m3 has the following view of Resource access membership matrix. Rows represents members m_1, m_2, m_3, m_5, m_6 and Columns represents Resources R_1, R_2.

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Member m3 has information of m5 and m6 because they are the members of resource group R2. RAM matrix maintained by m3 indicates that m1 is part of resource group R1, m2 is part of resource R2, m3 is part of resource R1 and R2, m4 is part of resource R2 and m5 is a part of resource R2.

11) Each member of R2 except m3 has the following view of Resource Membership Matrix. R1=m1, m2, m3. Rows represents members m_3, m_5, m_6 and Columns represents Resources R_1, R_2.

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Figure 2 shows that m1, m2 are the members of resource R1 and m5, m6 are the members of resource R2. Figure 3 shows member m3 joins R1. Figure 4 shows member m3 builds key graph as it is overlapped with R1 and R2. Figure 5 shows each Member of R1 and R2 key tree view.

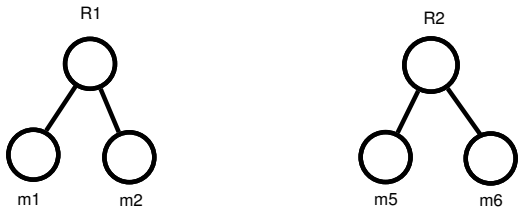


Figure 2: Members and resources

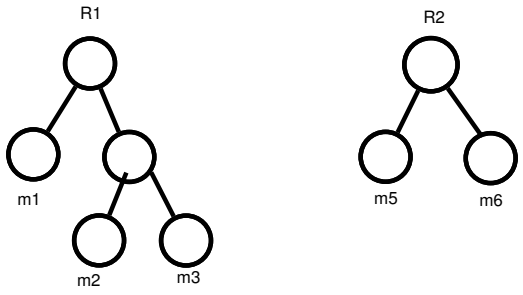


Figure 3: M3 joins R1

Table 1 illustrates the analysis of when single member joins to access the resource.

4.2 Batch Join

Buchade and Ingle states the algorithm when multiple tenants in a batch wants to access the resources [4]. The algorithm is classified into

1) Some tenants in a batch access single resource;
2) Some tenants in a batch access the multiple resources.

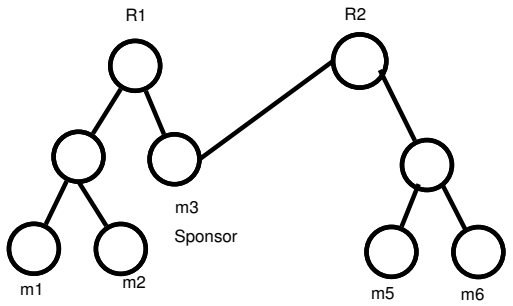


Figure 4: M3 joins R1 and R2, keygraph at M3

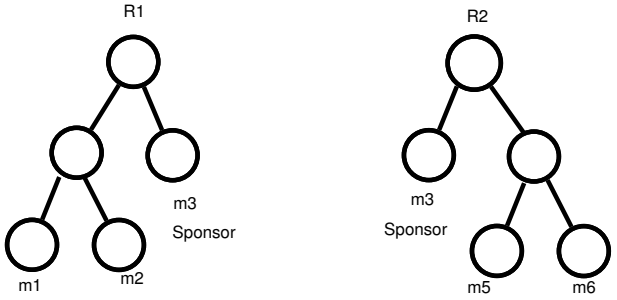


Figure 5: Each member view of R1 and R2

Key trees are build by considering overlapping members and non overlapping members in multiple resources. The details are given in the example.

4.2.1 Example: Batch Member Join

In existing key management algorithm [11, 16, 20, 24], seperate key tree is built for each resource, even if same members are accessing multiple resources.

For HDTV, Enhance layer channel subscribers can see enhance layer, Medium layer and Basic layer TV Channel. Medium layer channel subscribers can see Basic layer and Medium layer TV Channel. Basic layer channel subscribers can see Basic layer TV channel.

In existing approach, Enhanced layer members has to maintain three types of key trees.

1) For accessing EL Channel;
2) For accessing ML Channel;
3) For accessing BL Channel.

Our approach combines all key trees and eliminates redundant operations. Thus it helps to reduce key establishment time.

Each EL subscriber maintains resource access membership matrix. EL Channel considered as R1, ML Channel considered as R2 and BL Channel considered as R3. Any member broadcast request for joining the resource, the entry by EL subscriber is made in the resource membership matrix. EL subscriber forms key tree as mentioned in Figure 6.

Table 1: Analysis of single join

SE for SKT	$(N+1)(2\log(N+1)-2)$ where N = total Number of members
SE for CKT	if overlapped $(N)(2\log(N)-2)-(cm+1)(2\log(cm+1)-2)$ where cm = Number of overlapping members if not overlapped $(N+1)(2\log(N+1)-2)-(cm)(2\log(cm)-2)$

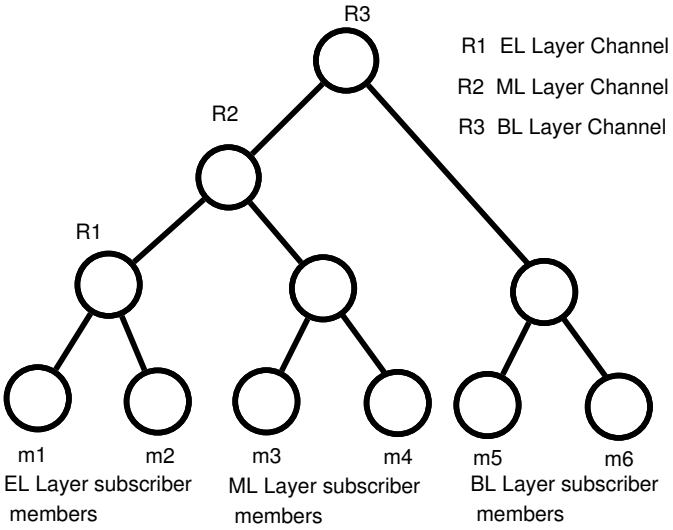


Figure 6: Key tree for EL members

EL members maintains the entries in resource access membership matrix is shown below. Members m1, m2, m3, m4, m5 and m6 represents the rows while Resources R1, R2 and R3 represents the entries in columns.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Each ML member maintains resource access membership matrix. ML Channel considered as R2 and BL Channel considered as R3. Any member broadcast request for joining the resource, the entry by ML subscriber is made in the resource membership matrix. ML subscriber forms key tree as mentioned in Figure7.

ML members maintains the entries in resource access membership matrix is shown below. Members m3, m4, m5 and m6 represents the rows while Resources R2 and R3 represents the entries in columns respectively.

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

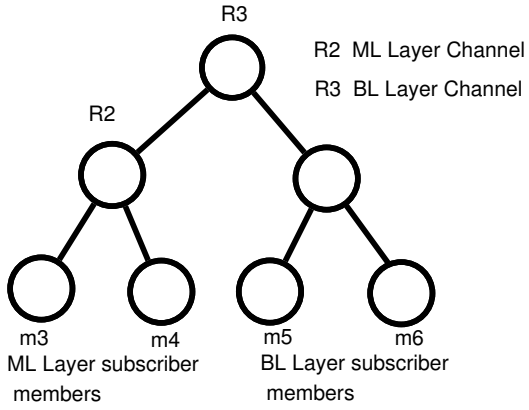


Figure 7: Key tree for ML members

Each BL member maintains resource access membership matrix. Any member broadcast request for joining the resource, the entry by BL subscriber is made in the resource membership matrix. BL subscriber forms key tree as mentioned in Figure 8.

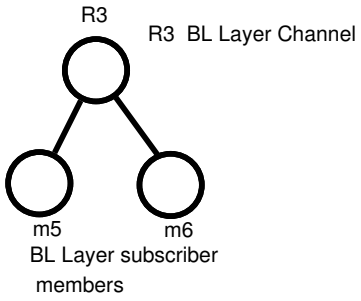


Figure 8: Key tree for BL members

BL members maintains the entries in resource access membership matrix is shown below. Members m5 and m6 represents the rows while Resources R1 represents the entries in columns.

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Thus our approach is more suitable for the applications mentioned in [9] and reduces computation overhead in terms of sequential exponential.

Table 2 illustrates the analysis when multiple members join to access the resources.

Table 2: Analysis of batch join

SE for SKT	$(N+n)(2\log(N+n)-2)$
SE for CKT	if all subscribers overlapped $(N)(2\log(N)-2)-(cm+n)(2\log(cm+n)-2)$ if some subscribers overlapped $(N+nom)(2\log(N)-2)-(cm+om)(2\log(cm+om)-2)$ nom = Number of not overlapping members om = Number of overlapping members if not overlapped $(N+n)(2\log(N+n)-2)-(cm)(2\log(cm)-2)$

4.3 Single Leave

Buchade and Ingle states the algorithm when member leaves the access of resource [4]. It broadcast the leave request. Entry is removed from the resource access membership matrix. Each member build the key tree by considering overlapping and non overlapping members. The Table 3 gives the analysis of single leave.

4.4 Batch Leave

Buchade and Ingle states the algorithm when members in a batch leaves the access of resources [4]. The entries of the same is made in the resource membership matrix. The entries of member removed from RAM matrix when members not accessing any resources. Overlapping members builds the key tree graph and non overlapping members builds the key tree. The analysis of the batch leave is given in Table 4.

5 Results and Analysis

Analysis is done by taking resources, varying group size and overlapping members.

From Figure 9, it is observed that when number of resources are 2, Number of overlapping members, cm=30 and group size varying, Number of sequential exponentiation required for separate resource key trees required more as compared to combined resource key trees.

From Figure 10, it is observed that when number of resources are 2, group size = 200 and overlapping members varying, Number of sequential exponentiation required for separate resource key trees required (23.66%) more as compared to combined resource key trees.

From Figure 11, it is observed that when group size = 200, overlapped members = 30 and as we varying the Number of resources, sequential exponential operations for separate resource key trees are more (11.16%) as compared to combined resource key trees.

From Figure 12, it is observed that when Number of resources are 2, overlapping members = 50 and group size varying, key establishment time required more for separate resource key trees as compared to combined resource key trees.

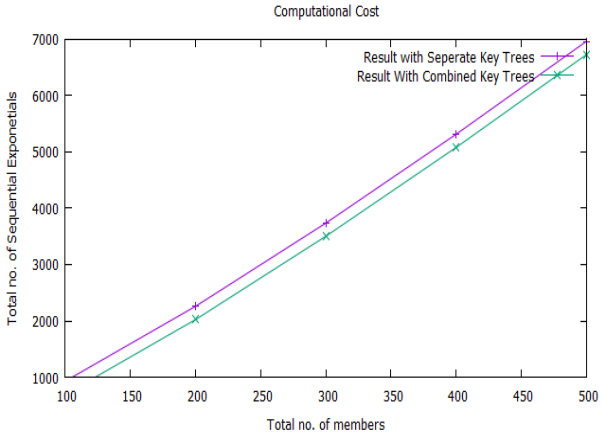


Figure 9: Computational cost, Number of resources = 2, cm = 30

From Figure 13, it is observed that when Number of resources are 2, group size = 100 and overlapping members varying key established time required for separate resource key trees required more as compared to combined resource key trees.

From Figure 14, it is observed that when Group size = 200, overlapped members = 50 and as we varying the Number of resources, key establishment time for separate resource key trees required more as compared to combined resource key trees.

6 Related Work

[14] proposes the scheme of tree key graph design but it has computation overhead for connection network generation. [2, 19] proposes share based key management scheme. KDC Scheme is used. It can cause single point of failure. Key-user tree is proposed. Storage cost is analyzed. Scheme is applicable to group communication. [8] proposes IGK scheme, considers TGDH approach. Author describes service group containing equal Number of members. But in real scenario, members can vary in the group. Sponsor selection is as per TGDH. The author ap-

Table 3: Analysis of single leave

SE for SKT	$(N-1)(2\log(N-1)-2)$
SE for CKT	if overlapping member leaves $(N)(2\log(N)-2)-(cm-1)(2\log(cm-1)-2)$ if non overlapped member leaves $(N-1)(2\log(N-1)-2)-(cm)(2\log(cm)-2)$

Table 4: Analysis of batch leave

SE for SKT	$(N-n)(2\log(N-n)-2)$ where N = Number of members
SE for CKT	if overlapping member leaves $(N)(2\log(N)-2)-(cm-om)(2\log(cm-om)-2)$ if non overlapped member leaves $(N-nom)(2\log(N-nom)-2)-(cm)(2\log(cm)-2)$ if non overlapping and overlapping member leaves $(N-nom)(2\log(N-nom)-2)-(cm-om)(2\log(cm-om)-2)$ where nom = non overlapping members and om = overlapping members

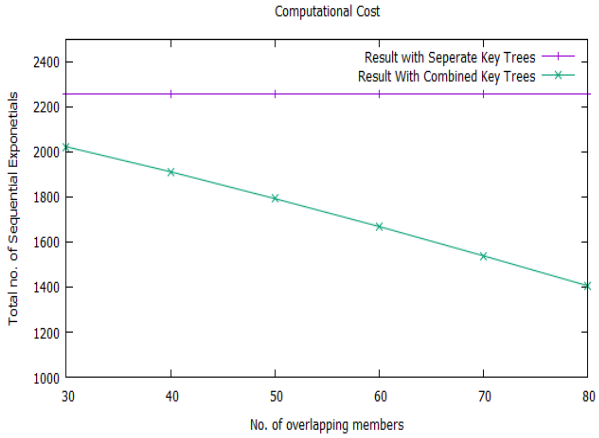


Figure 10: Average computational cost, Number of resources = 2, group size = 200

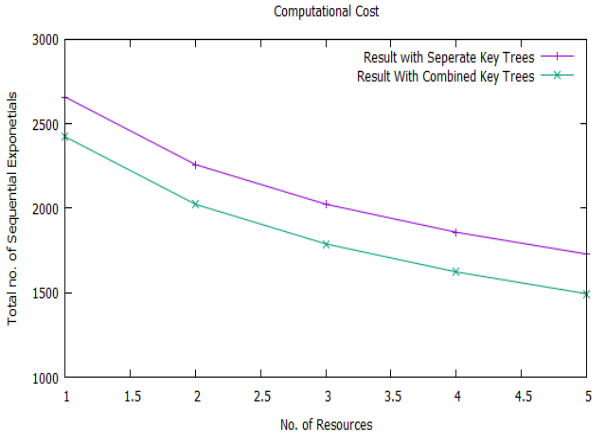


Figure 11: Average computational cost, group size = 200, overlapping members = 30

plies the scheme to specific type of example. [10] proposes tunable group key agreement protocol. Tree structure is used to form the group key among the members. [5] proposes share based hierarchical access control scheme is used. Group manager is considered. It assigns secret shares. Multi-group key management scheme is proposed. Computational analysis not done. [9] shows study of existing access control models done. Detailed analysis not done. [18] describes that group members are arranged in the hierarchical fashion. DH key agreement is applied. Sponsor not broadcasting blinded keys. But overlapping members not considered.

[16, 17] described group key formation techniques. It allows group members to consent on a shared group key. It is used to protect a shared file system present in the cloud. Any member can be sponsor. Concept of key lock boxes are used and represented in tree manner. Multiple members overlapping among different resources (e.g. files) in terms of group key management not considered. [7, 15, 22, 24, 25] uses TGDH but does not addresses issues of overlapping members. In [7] Huffman-based join-exit-tree scheme for contributory key management is proposed. It mainly concerns with key establishment time. But it does not concern with overlapping members. [3] describes key management methods and how it can be applied to computing scenario. Group key management method is also mentioned but not in detail. [11] Decisional square Diffie hellman approach is used. [21] proposes group key agreement protocol. Computational Diffie-Hellman used. But does not consider overlapping members.

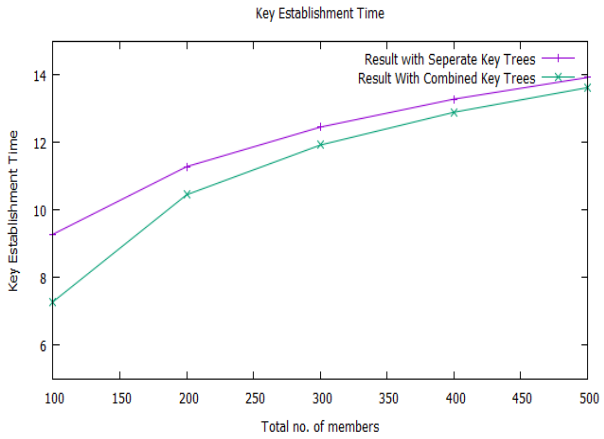


Figure 12: Key establishment time, Number of resources = 2, cm = 50

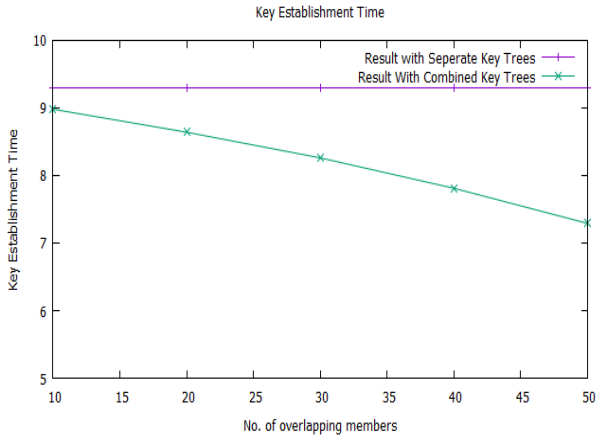


Figure 13: Key establishment time, Number of resources = 2, group size = 100

7 Conclusions

Group key is used to secure the access of resource in Cloud Computing. Group key is formed by tenants using resource key tree. TGDH is used to form the group key by building the key tree. In existing scenario, different key trees are formed even if tenants are common in multiple groups to access the resources. It causes computational overhead. We have proposed advance TGDH in which key trees may be combined if there are overlapping members in groups. Examples and analysis of algorithms are given. Through the analysis it is observed that computational overhead with respect to sequential exponentiation operations is decreased by 24% if we combine the key trees than the separate key trees. It is also observed that key establishment time for combined key trees is less compared to separate key trees.

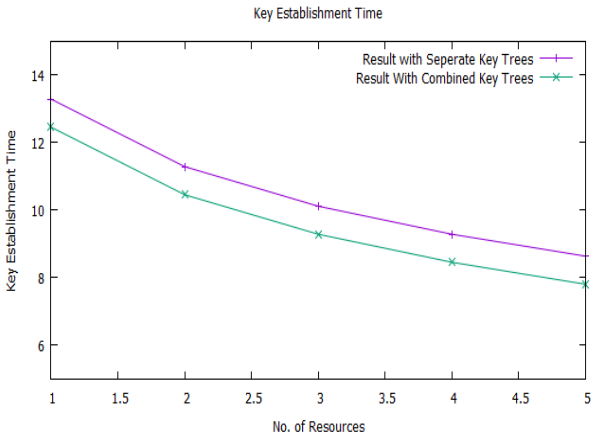


Figure 14: Key establishment time, group size = 200, overlapping members = 50

References

[1] J. Alves-Foss, "An efficient secure authenticated group key exchange algorithm for large and dynamic groups," in *In Proceedings of the 23rd National Information Systems Security Conference*, pp. 254–266, 2000.

[2] R. Aparna and B. B. Amberker, "Key management scheme for multiple simultaneous secure group communication," in *IEEE International Conference on Internet Multimedia Services Architecture and Applications*, pp. 1–6, 2009.

[3] A. Buchade and R. Ingle, "Key management for cloud data storage: Methods and comparisons," in *IEEE Fourth International Conference on Advanced Computing & Communication Technologies*, pp. 263–270, 2014.

[4] A. Buchade and R. Ingle, "Key trees combining algorithm for overlapping resource access members," *International Journal of Network Security*, vol. 18, no. 5, pp. 855–860, 2016.

[5] S. D. Dexter, R. Belostotskiy and A. M. Eskicioglu, "Multilayer multicast key management with threshold cryptography," in *Electronic Imaging*, pp. 705–715, International Society for Optics and Photonics, 2004.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[7] X. Gu, J. Yang, J. Lan and Z. Cao, "Huffman-based join-exit-tree scheme for contributory key management," *Computers & Security*, vol. 28, no. 1, pp. 29–39, 2009.

[8] X. Gu, Y. Zhao and J. Yang, "Reducing rekeying time using an integrated group key agreement scheme," *Journal of Communications and Networks*, vol. 14, no. 4, pp. 418–428, 2012.

[9] H. R. Hassen, A. Bouabdallah, H. Bettahar and Y. Challal, "Key management for content access con-

trol in a hierarchy," *Computer Networks*, vol. 51, no. 11, pp. 3197–3219, 2007.

[10] R. Ingle and G. Sivakumar, "Tunable group key agreement," in *32nd IEEE Conference on Local Computer Networks*, pp. 1017–1024, 2007.

[11] S. Jarecki, J. Kim and G. Tsudik, "Flexible robust group key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 879–886, 2011.

[12] D. H. Je, J. S. Lee, Y. Park and S. W. Seo, "Computation-and-storage-efficient key tree management protocol for secure multicast communications," *Computer Communications*, vol. 33, no. 2, pp. 136–148, 2010.

[13] Y. Kim, A. Perrig and G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 60–96, 2004.

[14] H. S. Koo, O. Kwon, S. W. Ra, et al. "A tree key graph design scheme for hierarchical multi-group access control," *IEEE Communications Letters*, vol. 13, no. 11, pp. 874–876, 2009.

[15] K. Kumar, V. Sumathy, et al. "A novel approach towards cost effective region-based group key agreement protocol for secure group communication," *arXiv preprint arXiv: 1007.0087*, 2010.

[16] I. Lam, S. Szebeni and L. Buttyán, "Invitation-oriented tgdh: Key management for dynamic groups in an asynchronous communication model," in *IEEE 41st International Conference on Parallel Processing Workshops*, pp. 269–276, 2012.

[17] I. Lam, S. Szebeni and L. Buttyán, "Tresorium: cryptographic file system for dynamic groups over untrusted cloud storage," in *IEEE 41st International Conference on Parallel Processing Workshops*, pp. 296–303, 2012.

[18] S. A. Mortazavi, A. N. Pour and T. Kato, "An efficient distributed group key management using hierarchical approach with diffie-hellman and symmetric algorithm: Dhsa," in *IEEE International Symposium on Computer Networks and Distributed Systems*, pp. 49–54, 2011.

[19] B. R. Purushothama and B. B. Amberker, "Group key management scheme for simultaneous multiple groups with overlapped membership," in *IEEE Third International Conference on Communication Systems and Networks*, pp. 1–10, 2011.

[20] M. Rajaram and D. Thilagavathy, "An interval based contributory key agreement," in *IEEE International Conference on Wireless Communication and Sensor Computing*, pp. 1–6, 2010.

[21] R. S. Ranjani, D. L. Bhaskari and P. S. Avadhani, "An extended identity based authenticated asymmetric group key agreement protocol," *International Journal of Network Security*, vol. 17, no. 5, pp. 510–516, 2015.

[22] Y. Sun and K. J. Liu, "Hierarchical group access control for secure multicast communications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1514–1526, 2007.

[23] G. Wang, O. Jie, H. H. Chen and M. Guo, "Efficient group key management for multi-privileged groups," *Computer Communications*, vol. 30, no. 11, pp. 2497–2509, 2007.

[24] H. Xiong, X. Zhang, W. Zhu and D. Yao, "Cloudseal: End-to-end content protection in cloud-based storage and delivery services," in *Security and Privacy in Communication Networks*, pp. 491–500, Springer, 2012.

[25] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 459–470, 2014.

Rajesh Ingle is adjunct professor at Department of Computer Engineering, Government College of Engineering Pune. He is professor in Department of Computer Engineering, Pune Institute of Computer Technology, Pune. He has received Ph.D. CSE from Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai. He has received the B.E. Computer Engineering from Savitribai Phule University of Pune, and M.E. Computer Engineering from Government College of Engineering, Savitribai Phule Pune University. He has also received M.S. Software Systems from BITS, Pilani, India. He is a senior member of the IEEE, IEEE Communications Society, and IEEE Computer Society. His research area is distributed system security, grid middleware, cloud security, multimedia networks and spontaneously networked environments.

Amar Buchade is research scholar at College of Engineering, Pune. He has received B.E. and M.E. in Computer Engineering from Walchand College of Engineering, Sangli in 2002 and 2005 respectively. His research area is distributed system, cloud computing and security.

On the CLD Attack to a Statistical Model of a Key Stream Generator

Shaoquan Jiang, Zailiang Tang, and Mingsheng Wang
(Corresponding author: Shaoquan Jiang)

Institute of Information Security, Mianyang Normal University
166 Mianxing Rd. West, High-Tech District, Mianyang 621000, China
(Email: shaoquan.jiang@gmail.com)
(Received Apr. 16, 2015; accepted Dec. 16, 2015)

Abstract

An embedding attack based on constraint Lenvenshtein distance was proposed by Golić and Mihaljević to analyze a statistical model of a key stream generator which contains an additive noise of probability p, where any value of $p < 1/2$ is possible. This attack is significant only if the embedding error caused by the noise is less than that caused by an incorrect candidate initial state. We show that this condition is not satisfied when $p \geq 1/4$.

Keywords: Cryptanalysis, LFSR, probabilistic analysis, stream cipher

1 Introduction

A key stream generator is a function that maps a short key into a long stream, which can be used to efficiently encrypt a plaintext stream by bit-wise XORing with the latter. The main purpose is to make it fast. The lightweight key stream generator Sprout [1] is such an example. However, proposing a secure key stream generator is very tricky. In fact, Sprout has been effectively attacked [12, 17]. See [2, 4, 5] for other examples of key stream generators. In this paper, we consider the key stream generator based on a linear feedback shift register (LFSR) sequence [10] which is a very efficient mechanism for a key stream generator (of course, LFSR has many applications such as frequency-hopping communication [3]).

In fact, most of generators in the literature are designed using it. However, many of them are broken by exploiting the linearity of LFSRs; see the correlation attack [14] for an example. A popular method for this type of attack is to reduce a complicated generator to a statistical model $Y_i = X_i + E_i, i = 1, 2, \cdots$, where $\{X_i\}_{i\geq 1}$ is a secret LFSR sequence, $\{Y_i\}_{i\geq 1}$ is the key stream and E_i is a binary noise with $P(E_i = 1) = p < 1/2$. For instance, Zeng et al. [15] reduced generators [6] to this model and completely broke them using a linear syndrome attack.

Generators subject to this attack usually have a common feature: Its input LFSR is regularly clocked. An irregularly clocked key stream generator is desired. Golić et al. [7] studied the security of this type of generator by considering the model $Y_i = X_{f(i)} + E_i, i = 1, 2, \cdots$, where $P(E_i = 1) = p < 1/2$, $f(i) = i + \sum_{j=1}^{i} a_j$ and $\{a_j\}$ is another LFSR. They proposed a constrained embedding attack to this model. When $p = 0$, this model degenerates to a decimation generator.

Golić and O'Connor [8] proposed an (un)constrained embedding attack to this generator when the irregularly clocking step is bounded by D. The embedding probability for $D = 2$ was given in [9]. Zhang [16] proposed a new attack to the decimation generator.

Given a partial key stream $Y_1, \cdots, Y_n$ of the model $Y_i = X_{f(i)} + E_i$ with $\Pr(E_i = 1) = p < 1/2$, Golić and Mihaljević considered a noisy embedding attack: Try to embed Y^n into the prefix $\hat{X}_1, \cdots, \hat{X}_{2n}$ of a candidate LFSR $\hat{X}$ for X (assume the resulting error sequence is $\hat{E}^n$) and find $\hat{X}$ with the least $\sum_{i=1}^{n} \hat{E}_i$ as the solution for X. The attack succeeds if it gives the solution $\hat{X} = X$. Notice that when generating Y^n from X^{2n}, the noise sequence is $E_1, \cdots, E_n$. Hence, this attack is significant only if $\sum_{i=1}^{n} E_i < \sum_{i=1}^{n} \hat{E}_i$ for any LFSR $\hat{X}$ other than X (otherwise, $Y_1, \cdots, Y_n$ is less noisy when considered as generated from a wrong LFSR $\hat{X}$). In this paper, we show that this condition is invalid when $p \geq 1/4$.

2 Preliminaries

Notions: We will use the following notions.

- For a set $\mathcal{S}$, $s \leftarrow \mathcal{S}$ samples an element s from $\mathcal{S}$ uniformly randomly.
- For $j \leq n$, u_j^n denotes sequence $u_j, u_{j+1}, \cdots, u_n$, and sequence u_1^n is simply denoted by u^n.
- For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \cdots, n\}$.
- i.i.d. is a well-known abbreviation of "*independently identically distributed*".

2.1 LFSR and Key Stream Generator

A binary *linear shift register sequence* (LFSR) $S = s_0, s_1, \cdots$ is a sequence generated using a linear recursive relation $s_{j+k} = s_j c_{k-1} + s_{j+1}c_{k-2} + \cdots + s_{j+k-1}c_0$ over $\mathbb{F}_2$, starting with an *initial state* $(s_0, \cdots, s_{k-1})$, where $c_0, \cdots, c_{k-1} \in \mathbb{F}_2$ are called *connection coefficients*.

A *key stream generator* is a function $f : \{0,1\}^k \to \{0,1\}^*$ that maps a secret key $\mathbf{w} \in \{0,1\}^k$ into a long binary stream $z_1, z_2, \cdots$. It can be used to encrypt a plaintext stream $m_1, m_2, \cdots$ by simply bit-wise XORing: $m_1 \oplus z_1, m_2 \oplus z_2, \cdots$. When a receiver with the secret key $\mathbf{w}$, receives the ciphertext, he can recover the plaintext in an obvious way. For the generator to be useful, we must make sure it is secure against some attacks.

A relatively weak attack is a *ciphertext-only attack*, which requires an adversary to recover the secret key $\mathbf{w}$ when only a partial ciphertext stream is given. A widely considered attack is a *known plaintext attack*: The adversary is given a partial ciphertext and its corresponding plaintext and his objective is to recover the secret $\mathbf{w}$. Equivalently, the attacker is given a partial key stream $z_1, \cdots, z_n$, from which he tries to recover the secret $\mathbf{w}$. It is well-known from Berlekamp-Massey algorithm [13] that LFSR with an initial state and connection coefficients as the secret key is not a secure key stream generator. However, LFSR is a very useful tool to construct a reasonably secure key stream generator.

2.2 Hoeffding Inequality

We now introduce the famous Hoeffding inequality. For details, see [11].

Lemma 1. *[Hoeffding] Let $X_1, \cdots, X_n$ be n independent RVs with $a_i \le X_i \le b_i$ for $i = 1, \cdots, n$. Then, for $\forall t > 0$,*

$$P\Big(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \ge t\Big) \le e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}},$$

$$P\Big(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \le -t\Big) \le e^{-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}},$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(X_i)$.

3 Problem Statement

3.1 A Statistical Model of a Clock-controlled Generator

Golić and Mihaljević [7] considered the following statistical model of a key stream generator. Let $X = \{x_i\}_{i\ge1}$ and $A = \{a_i\}_{i\ge1}$ be two LFSR sequences. The key stream $Z = \{z_i\}_{i\ge1}$ is generated noisily as follows.

$$z_i = x_{f(i)} + e_i, i = 1, 2, \cdots, \quad (1)$$

where $f(i) = i + \sum_{t=1}^{i} a_t$ and $e_1, e_2, \cdots$, are i.i.d. with $P(e_i = 1) = p < 0.5$. Strictly, this is not a key stream generator as it involves an additive noise e_i. However, this could be a useful abstraction of a key stream generator. Specifically, e_i could be an a statistical approximation to a complicated structure.

The initial state recovering problem for this generator is to find the initial states of X and A, assuming the noise probability p, a partial key stream z^n and connection coefficients of X and A are known. A näive approach for this is to search for all possible initial states of X and A and verify whether $x_{f(i)}$ matches with z_i for $i = 1, \cdots, n$ with probability roughly p. However, if each of X and A has an initialize state length k, then it requires $O(2^{2k})$ times of tests.

3.2 Constrained Levenshtein Distance Attack

Golić and Mihaljević [7] proposed a noisy embedding attack based on *Constrained Levenshtein Distance* (CLD) to recover $\{x_i\}_{i\ge1}$ from z^n. We call it a CLD *attack*. Their approach is as follows. For each candidate $\hat{X}$ of X, they generate a partial sequence $\hat{x}^{2n}$ and compute the CLD between $\hat{x}^{2n}$ and the known key stream z^n, where CLD is defined as follows.

$D^*(\hat{x}^{2n}, z^n)$ = minimal number of deleting and complementation operations required to produce z^n from $\hat{x}^{2n}$, by first deleting an arbitrary prefix of $\hat{x}^m$ and then following the model at Equation (1).

Let $\mathcal{X}$ be the set of candidate $\hat{X}$ for X. If the initial state of X has k bits, then $|\mathcal{X}| = 2^k$. Given z^n, the attack outputs X^* that minimizes CLD (among all possible sequences in $\mathcal{X}$) as its solution for X. It succeeds if $X^* = X$.

For each $\hat{X}$, [7] showed that CLD$(\hat{x}^m, z^n)$ can be computed in $O(mn)$ time and hence is efficient.

Note that the number of deletions in producing z^n from $\hat{x}^{2n}$ is the constant n. This attack is equivalent to minimize the number of complementing operations. We denote the number of complementing operations in $D^*(\hat{x}^{2n}, z^n)$ by $D(\hat{x}^{2n}, z^n)$. In the sequel, instead of $D^*(\hat{x}^{2n}, z^n)$, we will focus on $D(\hat{x}^{2n}, z^n)$.

Note that z^n is generated from X, A in the real process with the complementing sequence e^n. So the number of complementing operations in this process is $\sum_{i=1}^{n} e_i$. Hence, the CLD attack is meaningful only if $\sum_{i=1}^{n} e_i < D(\hat{x}^{2n}, z^n)$ for each $\hat{X} \in \mathcal{X} - \{X\}$. That is, the real number of complementing operations should not be greater than that under a wrong candidate sequence $\hat{X}$. Hence, we consider

$$\alpha = P\Big(D(\hat{x}^{2n}, z^n) \le \sum_{i=1}^{n} e_i\Big). \quad (2)$$

As in [7], we model an LFSR as a purely random sequence. So α is defined over the uniform random $\hat{x}^{2n}, x^{2n}$

and the randomness of z^n and e^n.

The expected number of $\hat{X}$ in $\mathcal{X}$ with $D(\hat{x}^{2n}, z^n) \leq \sum_{i=1}^{n} e_i$ is $2^k\alpha$, as $|\mathcal{X}| = 2^k$. Thus, the CLD attack is meaningful only if $2^k\alpha$ is small. The problem in this paper is to lower bound α and show that $\alpha > constant$ when $p \geq 1/4$. In this case, $2^k\alpha$ is large, which makes the attack fail to identify which $\hat{X}$ will be the true X.

Finally, we notice that $D^*(\hat{x}^{2n}, z^n)$ permits deleting an arbitrary *prefix* of $\hat{x}^m$. However, here the prefix can be changed to postfix without affecting α in Equation (2). Indeed, we can convert a postfix into a prefix version in the following way. We can start to embed z^n reversely to $\hat{x}^{2n}$. That is, we can embed $z_n, z_{n-1}, \cdots, z_1$ into $\hat{x}_{2n}, \cdots, \hat{x}_1$. If z_1 is embedded at $\hat{x}_j$, then $\hat{x}^{j-1}$ can be deleted by the convention.

Since z^n and $\hat{x}^{2n}$ are uniformly random and independent, the distribution of $z_n, \cdots, z_1, \hat{x}_{2n}, \cdots, \hat{x}_1$ and the distribution of $z_1, \cdots, z_n, \hat{x}_1, \cdots, \hat{x}_{2n}$ are exactly the same. So the two ways give the same α. In this paper, for convenience, we use the postfix version for $D(\hat{x}^{2n}, z^n)$ (i.e., we revise "prefix" in the definition of $D^*(\hat{x}^{2n}, z^n)$ to "postfix").

4 Lower Bound on α When $p \geq 1/4$

In this section, we show that α is larger than a constant when $p \geq 1/4$. Our strategy is as follows. For an embedding algorithm $\mathcal{E}$ that embeds z^n into u^{2n}, we use $\mathcal{E}(u^{2n}, z^n)$ to denote the number of flips in the embedding process. Then, $\mathcal{E}(u^{2n}, z^n) \geq D(u^{2n}, z^n)$. It follows that $\alpha \geq P(\mathcal{E}(u^{2n}, z^n) \leq \sum_{i=1}^{n} e_i)$. Hence, it suffices to show that $P(\mathcal{E}(u^{2n}, z^n) \leq \sum_{i=1}^{n} e_i) > constant$ for some algorithm $\mathcal{E}$. So the main task is to design $\mathcal{E}$.

Now we present our algorithm $\mathcal{E}$ to embed z^n to u^{2n}, in which the average number of complements is $n/4$. The formal description is in Algorithm 1. The idea is as follows. It sequentially embeds $z_1, \cdots, z_n$ into u^{2n}. Let z_i be the current bit to be embedded and u_j be the currently unused bit awaiting to embed z_i. Initially, $i = j = 1$. If $z_i \neq u_j$ and $z_i \neq u_{j+1}$, then one complementing operation (recorded in a variable F) is used and z_i is embedded at u_{j+1}; otherwise, z_i is embedded to u_j when $z_i = u_j$ and embedded to u_{j+1} when $z_i = u_{j+1}$. Finally, increment i and update j to the next unused index.

Before analyzing our algorithm, we first present a general lemma. It considers a function $f : \{0,1\}^m \to [m-1]$. It states that if f satisfies a certain property, then for uniformly random U^m in $\{0,1\}^m$ and $J = f(U^m)$, we have that (U_J, U_{J+1}) is independent of U^{J-1}. We remark that this independency does not trivially follow from the uniformly randomness of U^m, as J depends on U^m and is implied from U^{J-1} (by looking at the dimension).

Lemma 2. *Let $f : \{0,1\}^m \to [m-1]$ be a function with the following property: if $f(u^m) = j$, then $f(u^{j-1}, v_j^m) = j$ for any $v_j^m \in \{0,1\}^{m-j+1}$. Let U^m be uniformly random in $\{0,1\}^m$ and $J = f(U^m)$. Then, (U_J, U_{J+1}) is independent of U^{J-1}.*

Algorithm 1 Embedding algorithm $\mathcal{E}$

Imput: u^{2n}, z^n;
Output: F
1: Begin
2: Set $i = 1, j = 1, F = 0$
3: **for** $i = 1$ to n **do**
4: **if** $z_i = u_j$ **then**
5: $j = j + 1$
6: **else**
7: **if** $z_i = u_{j+1}$ **then**
8: $j = j + 2$;
9: **else** $F = F + 1$ and $j = j + 2$;
10: **end if**
11: **end if**
12: **end for**
13: Return F
14: End

Proof. For $j \in [m-1]$, let $\mathcal{N}_j$ be the set of u^{j-1} such that $f(u^{j-1}, v_j^m) = j$ for some v_j^m. By the property of f, the set of all u^m with $f(u^m) = j$ is exactly $\mathcal{S}_j \stackrel{def}{=} \mathcal{N}_j \times \{0,1\}^{m-(j-1)}$. As any u^m must map to some $j \in [m-1]$, it follows that $\sum_{j=1}^{m-1} |\mathcal{N}_j| 2^{m-(j-1)} = 2^m$. So

$$\sum_{j=1}^{m-1} |\mathcal{N}_j| 2^{-(j-1)} = 1. \tag{3}$$

Notice that J can be derived from U^{J+1} by looking at the dimension. Hence, $U^{J+1} = u^{j+1}$ if and only if $J = j$ and $U^{j+1} = u^{j+1}$. So,

$$\begin{aligned} P(U^{J+1} = u^{j+1}) &= P(U^{j+1} = u^{j+1}, J = j) \\ &= P_{U^{j+1}}(u^{j+1}) P_{J|U^{j+1}}(j|u^{j+1}) \\ &= \begin{cases} 2^{-(j+1)}, & u^{j-1} \in \mathcal{N}_j \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where we have used the fact $P_{J|U^{j+1}}(j|u^{j+1}) = 1$ if $u^{j-1} \in \mathcal{N}_j$ and zero, otherwise. Similarly,

$$\begin{aligned} P(U^{J-1} = u^{j-1}) &= P(U^{j-1} = u^{j-1}, J = j) \\ &= P_{U^{j-1}}(u^{j-1}) P_{J|U^{j-1}}(j|u^{j-1}) \\ &= \begin{cases} 2^{-(j-1)}, & u^{j-1} \in \mathcal{N}_j \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

Therefore, $P_{U^{J+1}}(u^{j+1}) = \frac{1}{4} P_{U^{J-1}}(u^{j-1})$. As

$$\begin{aligned} P_{U_J U_{J+1}}(a,b) &= \sum_{j=1}^{n} P_{U_j U_{j+1} J}(a,b,j) \\ &\stackrel{(*)}{=} \sum_{j} |\mathcal{N}_j| \cdot 2^{-(j+1)} \\ &= 1/4, \text{ (by Equation (3))} \end{aligned} \tag{6}$$

where equality $(*)$ follows from the fact that $(U_j, U_{j+1}, J) = (a, b, j)$ if and only if

$$U^m \in \mathcal{N}_j \times \{a\} \times \{b\} \times \{0,1\}^{m-(j+1)},$$

and the fact that U^m is uniformly random over $\{0,1\}^m$. Hence, Equations (4) (5) (6) imply

$$P_{U^{J+1}}(u^{j+1}) = P_{U_J U_{J+1}}(u_j, u_{j+1}) P_{U^{J-1}}(u^{j-1}). \quad (7)$$

That is, (U_J, U_{J+1}) is independent of U^{J-1}. □

We are ready to analyze our algorithm $\mathcal{E}$. We will use the following notations. Let $U^{2n} = U_1, \cdots, U_{2n}$ be a sequence of purely random binary stream. We define a binary RV F_i with respect to algorithm $\mathcal{E}(U^{2n}, z^n)$ such that $F_i = 1$ if and only if $F = F+1$ is executed in loop i. Then, $F = \sum_{i=1}^n F_i$. Since $\mathcal{E}$ is deterministic, the randomness of F is over U^{2n}. Let J_i be the index j at the beginning of loop i (e.g., $J_1 = 1$). Define function $\delta : \{0,1\}^2 \to \{0,1\}$ is defined such that $\delta(x, y) = 1$ if and only if $x = y$.

In the following, we show that $F_1, \cdots, F_n$ are independent and that $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are independent too, where the independency for both collections follows only from the randomness of U^{2n}.

Lemma 3. *Given $z^n \in \{0,1\}^n$, if U^{2n} is uniformly random in $\{0,1\}^{2n}$, then*

1) *RVs $F_1, \cdots, F_n$ are i.i.d. with $P(F_i = 1) = 1/4$.*
2) *RVs $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are i.i.d. with $P(\delta(U_{J_i}, z_i) = 1) = 1/2$.*

Proof. Notice that for any $i \le n$, we have $J_i < 2n$. By Lemma 2, if $f(U^{2n}) = J_i$, then (U_{J_i}, U_{J_i+1}) is independent of U^{J_i-1}. This will be used in our proof.

1) We start with the following claim.

   Claim 1. *For fixed z^n and i, we have that $(F_1, \cdots, F_{i-1})$ is deterministic in U^{J_i-1}.*

   Proof. Indeed, by our algorithm, if $z_{i-1} = U_{J_{i-1}}$, then $J_i = 1 + J_{i-1}$ and $F_{i-1} = 0$; otherwise, $J_i = 2 + J_{i-1}$, and $F_{i-1} = 1$ if and only if $z_{i-1} \neq U_{1+J_{i-1}}$. Here we can see that in any case, F_{i-1} is computed only using U^{J_i-1}. So for any $\ell < i$, $F_{\ell-1}$ is determined by $U^{J_\ell-1}$ (which in turn is determined by U^{J_i-1}). Thus, $(F_1, \cdots, F_{i-1})$ are deterministic in U^{J_i-1}, when z^n is fixed. This concludes the proof of our claim. □

   From the algorithm description, we can write $F_i = (z_i \oplus U_{J_i}) \wedge (z_i \oplus U_{1+J_i})$. Thus, F_i is deterministic in (U_{J_i}, U_{1+J_i}). From claim 1 and the fact that (U_{J_i}, U_{J_i+1}) is independent of U^{J_i-1} (see the beginning of the proof), we know that F_i is independent of $(F_1, \cdots, F_{i-1})$.

   Finally, notice that (U_{J_i}, U_{1+J_i}) is independent of J_i, as J_i is deterministic in U^{J_i-1} (by looking at the dimension). Hence,

$$\begin{aligned} & P((U_{J_i}, U_{1+J_i}) = (a,b)) \\ = & \sum_j P((U_j, U_{j+1}, J_i) = (a,b,j)) \\ = & \sum_j P((U_j, U_{j+1}) = (a,b)) P(J_i = j) \\ = & \frac{1}{4} \sum_j P(J_i = j) = 1/4. \end{aligned}$$

   Hence, from $F_i = (z_i \oplus U_{J_i}) \wedge (z_i \oplus U_{1+J_i})$, we have $P(F_i = 1) = 1/4$.

2) As U_{J_i} is independent of U^{J_i-1}, we have $\delta(U_{J_i}, z_i)$ is independent of $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_{i-1}}, z_{i-1})$ for any i. Hence, $\delta(U_{J_1}, z_1), \cdots, \delta(U_{J_n}, z_n)$ are independent. Finally, as J_i is deterministic in U^{J_i-1} (by looking at the dimension), U_{J_i} is independent of J_i. Thus,

$$\begin{aligned} P(U_{J_i} = 0) &= \sum_j P((U_j, J_i) = (0,j)) \\ &= \sum_j P(U_j = 0) P(J_i = j) \\ &= \frac{1}{2} \sum_j P(J_i = j) = 1/2. \end{aligned}$$

   This completes our proof.

□

We are ready to prove our theorem. This mainly is achieved using Heoffding inequality to $F_1, \cdots, F_n$ and the true error sequence $e_1, \cdots, e_n$ in producing z^n.

Theorem 1. *If $p = 1/4$, then $\alpha \ge 1/2$; if $p > 1/4$, then $\alpha \ge 1 - e^{-(p-.25)^2 n}$.*

Proof. Notice that $F = \sum_{i=1}^n F_i$ is the number of complements in a specific embedding process. Hence, $F \ge D(U^{2n}, z^n)$. Hence,

$$\begin{aligned} \alpha =& P\Big(D(U^{2n}, z^n) \le \sum_{i=1}^n e_i\Big) \\ \ge& P\Big(\sum_{i=1}^n F_i \le \sum_{i=1}^n e_i\Big) \end{aligned} \quad (8)$$

Since F^n only depends on U^{2n}, it is independent of e^n. If $p = 1/4$, then $F_1, \cdots, F_n$ are identically distributed with $e_1, \cdots, e_n$. Then, by symmetry, $\alpha \ge 1/2$. If $p > 1/4$, then since $F_1, \cdots, F_n, e_1, \cdots, e_n$ are independent, by Hoeffding inequality with $2n$ random variables,

$$\begin{aligned} & P(e_1 + \cdots + e_n - F_1 - \cdots - F_n \ge 0) \\ =& 1 - P(\sum_i (e_i - F_i) - n(p - .25) < -n(p - .25)) \\ \ge& 1 - e^{-(p-.25)^2 n}. \end{aligned} \quad (9)$$

This completes our theorem. □

Now we look at how many bits Algorithm $\mathcal{E}$ has used in order to embed z^n. In fact, after embedding z_n, the next available index of U_j is J_{n+1}. So the number of bits in embedding z^n is $N_n \stackrel{def}{=} J_{n+1} - 1$. From our algorithm description, $J_{\ell+1} = 1 + \delta(U_{J_\ell}, z_\ell) + J_\ell$. Thus,

$$N_n = n + \sum_{i=1}^{n} \delta(U_{J_i}, z_i). \tag{10}$$

Notice that in the real process in producing z^n, we know that $f(n) = n + \sum_{i=1}^{n} a_i$, where $a_1, a_2, \cdots$, are i.i.d. and uniformly random over $\{0, 1\}$ (as idealized in our analysis). Therefore, by Lemma 3, the distribution of N_n is identical to the real distribution. This demonstrates an interesting aspect of our algorithm.

5 Conclusion

In this paper, we revisited the noisy embedding attack from constraint Lenvenshtein distance by Golić and Mihaljević to a noisy key stream generator that contains an additive binary noise term of probability p, where any value of $p < 1/2$ is possible. We showed that this attack is not successful if $p \geq 1/4$. One immediate interesting question is to study the success for the case $p < 1/4$. When p is very small, the exponentially small embedding probability without a noise showed in [9] trivially implies the success of this algorithm. However, in general, this does not seem to be a trivial question. We leave it as an open question.

Acknowledgements

This work is supported by Open grant (No. 2015-MS-11) of State Key Lab of Information Security, Institute of Information Engineering, CAS.

References

[1] F. Armknecht and V. Mikhalev, "On lightweight stream ciphers with shorter internal states," in *Fast Software Encryption*, pp. 451–470, Springer-Verlag, 2015.

[2] Y. Asimi, A. Amghar, A. Asimi and Y. Sadqi, "New random generator of a safe cryptographic salt per session," *International Journal of Network Security*, vol. 18, no. 3, pp. 445–453, 2016.

[3] J. Chung, G. Gong and K. Yang, "New families of optimal frequency-hopping sequences of composite lengths," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3688–3697, 2014.

[4] H. El-Razouk, A. Reyhani-Masoleh and G. Gong, "New implementations of the wg stream cipher," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1865–1878, 2014.

[5] H. El-Razouk, A. Reyhani-Masoleh and G. Gong, "New hardware implementations of WG (29, 11) and WG-16 stream ciphers using polynomial basis," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 2020–2035, 2015.

[6] P. R. Geffe, "How to protect data with ciphers that are really hard to break," *Electronics*, vol. 4, pp. 129–156, 1973.

[7] J. D. Golić and M. J. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the levenshtein distance," *Journal of Cryptology*, vol. 3, no. 3, pp. 201–212, 1991.

[8] J. D. Golić and L. O'Connor, "Embedding and probabilistic correlation attacks on clock-controlled shift registers," in *Proceedings of Advances in Cryptology (Eurocrypt'94)*, pp. 230–243, Springer-Verlag, 1994.

[9] J. D. Golić, "Constrained embedding probability for two binary strings," *SIAM Journal on Discrete Mathematics*, vol. 9, no. 3, pp. 360–364, 1996.

[10] S. W. Golomb, *Shift Register Sequences*, San Francisco: Holden-Day, Inc., 1967.

[11] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[12] V. Lallemand and M. Naya-Plasencia, "Cryptanalysis of full sprout," in *Advances in Cryptology (Crypto'15)*, pp. 663–682, Springer-Verlag, 2015.

[13] J. L. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[14] T. Siegenthaler, "Decrypting a class of stream cipher using ciphertext only," *IEEE Transactions on Computers*, vol. 34, no. 1, pp. 81–85, 1985.

[15] K. Zeng and M. Huang, "On the linear syndrome method in cryptoanalysis," in *Proceedings of Advances in Cryptology (Crypto'88)*, pp. 469–478, Springer-Verlag, 1990.

[16] B. Zhang, "New cryptanalysis of irregularly decimated stream ciphers," in *Proceedings of 16th Annual International Workshop of Selected Areas in Cryptography*, pp. 449–465, Springer-Verlag, 2009.

[17] B. Zhang and X. Gong, "Another tradeoff attack on sprout-like stream ciphers," in *Advances in Cryptology (Asiacrypt'15)*, pp. 561–585, Springer-Verlag, 2015.

Shaoquan Jiang received the B.S. and M.S. degrees in mathematics from the University of Science and Technology of China, Hefei, China, in 1996 and 1999, respectively. He received the Ph.D degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, ON, Canada, in 2005. From 1999 to 2000, he was a research assistant at the Institute of Software, Chinese Academy of Sciences, Beijing; from 2005 to 2013, he was a faculty member at the University of Electronic Science and Technology of China, Chengdu, China; from 2013 to now, he is a faculty member at Mianyang Normal University, Mianyang, China. He was a postdoc at the

University of Calgary from 2006 to 2008 and a visiting research fellow at Nanyang Technological University from Oct. 2008 to Feb. 2009. His research interests are key stream generators, public-key based secure systems and secure protocols.

Insecurity of a Certificate-free Ad Hoc Anonymous Authentication

Yan Xu[1], Liusheng Huang[2], Miaomiao Tian[2], and Hong Zhong[1]
(Corresponding author: Hong Zhong)
School of Computer Science and Technology, Anhui University[1]
Hefei 230027, China
School of Computer Science and Technology, University of Science and Technology of China[2]
Hefei 230601, China
(Email: xuyan@ahu.edu.cn)
(Received Apr. 14, 2015; revised and accepted July 30 & Nov. 27, 2015)

Abstract

The ring signature scheme is a simplified group signature scheme for no manager while preserving unconditionally anonymous of the signer. Certificateless cryptography is introduced for eliminating the use of certificates in Public Key Infrastructure and solving the key-escrow problem in ID-based cryptogratography. Recently, Qin et al. proposed the first RSA-based certificateless ring signature scheme which was proved unforgeable in random oracle model. In this paper, we demonstrated that this scheme was not secure against the Type I adversary.

Keywords: Certificateless cryptography, ring signature, RSA

1 Introduction

In 2001, Rivest et al. [11] formally introduced the concept of the ring signature in which the verifier can be convinced that the message was authenticated by a ring including the signer while keeping the signer unconditionally anonymous. Anonymity and spontaneity are inherent properties of the ring signature. Anonymity allows anyone to verify the validity of the ring signature without revealing the signer's identity. Spontaneity means that the signer can generate the ring signature without any help or cooperation from the other ring members. The ring signature allows the signer to decide all ring members. The ring signature scheme in [11] is based on RSA cryptosystem. Abe et al. [1] proposed the first ring signature scheme based on discrete logarithm problem. These ring signature schemes are all based on traditional Public Key Infrastructure which requires a great amount of computing time and storage to manage the certificates. In order to avoid the heavy burden of certificate management, Shamir [12] introduced Identity-based public key cryptography (ID-PKC). In 2002, Zhang et al. [16] proposed the first ID-based ring signature scheme. Nguyen [9] proposed the first ring signature with a constant number of pairing computations and a constant size signature. Au et al. [3] proposed the first secure ring signature scheme in standard model. Herranz [7] and Tsang et al. [14] respectively provided the ID-based ring signature schemes from RSA. However, ID-based cryptography usually suffers from the inherent key escrow problem.

In 2003, Al-Riyami and Paterson [2] introduced the concept of certificateless public key cryptography (CL-PKC) which not only avoids the key escrow problem but also moves the digital certificates. In CL-PKC, there is a third party called Key Generate Center (KGC) to issue the users partial private keys with their identities. However, the KGC has no right to access the full private key which is generated by combining the partial private key and a secret value chosen by the user itself. The public keys are computed by the secret value and then published by users. The CL-PKC has attracted a lot of further studies [6, 8, 13]. Yum et al. [15] proposed a general construction of certificateless signature (CLS) scheme which was a less efficient scheme. Zhang and Mao [17] designed the first RSA-based CLS scheme.

In 2007, two certificatelss ring signature (CL-RS) schemes [5, 18] were proposed independently. Chang et al. [4] constructed a more efficient (t,n) threshold ring signture scheme. The above CL-RS schemes are all based on bilinear pairings which is an expensive operation for the computational cost. Qin et al. [10] proposed the first RSA-based CL-RS scheme without bilinear parings and proved their scheme was secure in random oracle model. However, we found that Qin et al.'s scheme was vulnerable to a Type I adversary who can replace the public key of any signer.

2 Preliminaries

2.1 Security Model of the Certificateless Ring Signature Scheme

There are two kinds of adversaries in the security model of CL-RS scheme. Type I adversary $\mathcal{A}_1$ can replace the public key of any user at his will but is not able to visit the partial private key. Type II adversary $\mathcal{A}_2$ models the malicious-but-passive KGC who generates the partial private keys for users, but cannot replace any users' public keys. We define two games, **Game 1** for $\mathcal{A}_1$, and **Game 2** for $\mathcal{A}_2$.

- **Game 1**: Let S_1 be the challenger to interactive with $\mathcal{A}_1$

  1) **Initialization:** S_1 runs **Setup** and **MasterKeyGen** algorithms to get the system parameters mpk and the master key pair msk. Then S_1 publics mpk while keeping msk secret. S_1 maintains three lists L_1, L_2, L_3 initiated empty. (1) L_1 records the identities whose partial private keys have been required by $\mathcal{A}_1$ in **PartialKeyGen** queries. (2)L_2 records the identities whose public keys have been replaced by $\mathcal{A}_1$. (3) L_3 records the identities who have been corrupted by $\mathcal{A}_1$ in **Corruption** queries.
  2) **Query:** $\mathcal{A}_1$ adaptively performs a polynomially bounded number of queries.
     - **UserKeyGen:** On input a user's identity ID, if ID has not been created, S_1 run **UserKeyGen** to generate (upk_{ID}, usk_{ID}), upk_{ID} is returned.
     - **PartialKeyGen:** $\mathcal{A}_1$ requests the partial private key of the user ID. If $ID \notin L_1$, S_1 first sets $L_1 = L_1 \cup ID$ and then runs **PartialKeyGen**. Otherwise S_1 does nothing. Finally psk_{ID} is returned.
     - **ReplaceKey**: On input ID and upk^*_{ID}, if ID has been requested in **UserKeyGen**, S_1 first sets $L_2 = L_2 \cup ID$ and then updates the public key of ID as upk^*_{ID}. Otherwise nothing is carried out.
     - **Corruption**: $\mathcal{A}_1$ requests the full private key of the user with identity ID.
       a. If $ID \in L_2$, S_1 cannot output the full private key of ID whose public key is replaced, S_1 returns $\perp$.
       b. Otherwise, S_1 first sets $L_3 = L_3 \cup ID$, and then returns the partial private key psk_{ID} as well as the user secret value usk_{ID}.
     - **Ring-Sign:** On input a message m, a ring R containing the identities and the public keys of ring members, S_1 outputs a ring signature σ.
  3) **Forgery:** At the end of the simulation, $\mathcal{A}_1$ outputs (R^*, m^*, σ^*) as the forgery. We say that $\mathcal{A}_1$ wins the game:
     - (R^*, m^*) has never been required for the verification.
     - $Verify(R^*, m^*, \sigma^*) = 1$ and $(L^*_{ID} \cap L_1 \cap L_2) \cup (L^*_{ID} \cap L_3) = \emptyset$ for L^*_{ID} is the set of ring members' identities.

- **Game 2**: Let S_2 be the challenger to interactive with $\mathcal{A}_2$

  1) **Initialization:** As with the initialization of **Game 1**, except that S_2 sends the master key pair (mpk, msk) to $\mathcal{A}_2$. In **Game 2**, lists L_2, L_3 are maintained by S_2.
  2) **Query:** $\mathcal{A}_2$ makes the queries of **UserKeyGen**, **Corruption** and **Ring-Sign** in the same way as in **Game 1**.
  3) **Forgery:** At the end of the simulation, $\mathcal{A}_2$ outputs (R^*, m^*, σ^*) as the forgery. We say that $\mathcal{A}_2$ wins the game:
     - (R^*, m^*) has never been required for the verification $Verify(R^*, m^*, \sigma^*) = 1$
     - $L^*_{ID} \cap L_3 = \emptyset$ for L^*_{ID} is the set of ring members' identities.

Definition 1. (Unforgeability). *A CL-RS scheme is unforgeable if the advantage of any polynimail bounded adversary in the **Game 1** and **Game 2** is negligible.*

3 Cryptanalysis of Qin *et al.* CL-RS Scheme

3.1 The Qin *et al.* 's CL-RS Scheme

- **Setup:** On input 1^k as a security parameter, the KGC randomly selects two k-bit prime number p, q and computes $N = pq$. The KGC picks two prime numbers e, d satisfying $\gcd(e, \varphi(n)) = 1$ and $ed = 1 \bmod \varphi(n)$, where $\varphi(n)$ denotes the Euler totient function. Finally, the KGC chooses two hash functions H_1, H_2 which satisfy $H_1 : \{0,1\}^* \to Z^*_N$ and $H_2 : \{0,1\}^* \to \{0,1\}^l$. The KGC publishes the public parameters $mpk = \{N, e, H_1, H_2\}$ while keeping the master key $msk = \{p, q, d\}$ secret.

- **PartialKeyGen:** For the user with $ID \in \{0,1\}^*$, the KGC computes its partial private key $psk_{ID} = H_1(ID)^d$.

- **UserKeyGen:** The user ID selects $x_{ID} \in Z_{2^{|N|/2-1}}$ as its secret value usk_{ID} and sets its public key $upk_{ID} = H_1(ID)^{x_{ID}}$, where $|N|$ denotes the binary length of N.

- **Ring-Sign:** Let $R = L_{ID} \cup L_{upk}$, $L_{ID} = \{ID_1, \cdots, ID_n\}$ denotes the set of ring members' identities with the corresponding set of public keys $L_{upk} = \{upk_{ID1}, \cdots, upk_{IDn}\}$. To sign a message $m \in \{0,1\}^*$ on behalf of the ring, the signer ID_π performs the following steps by using its full private key $SK_{ID\pi} = (psk_{ID\pi}, usk_{ID\pi})$.
  - Selects two random numbers $r_{\pi 1}$,$r_{\pi 2} \in Z_{2^{|N|/2-1}}$.
  - Computes $R_{\pi 1} = H_1(ID_\pi)^{r_{\pi 1}} \bmod N, R_{\pi 2} = H_1(ID_\pi)^{r_{\pi 2}} \bmod N$.
  - Randomly chooses $u_{i1}, c_i \in Z_N^*, u_{i2} \in Z_{2^{|N|/2-1}}$ pairwise different, for $i \in [1,n], i \neq \pi$. Then ID_π computes $R_{i1} = u_{i1}^e H_1(ID_i)^{c_i} \bmod N, R_{i2} = H_1(ID_i)^{u_{i2}} upk_{IDi}^{c_i} \bmod N$.
  - Computes $c_0 = H_2(m||L_{ID}||L_{upk}||(R_{i1}, R_{i2})i \in [1,n])$.
  - Generates a polynomial f over $GF(2^k)$ with degree $n-1$ such that $c_0 = f(0), c_i = f(i)$ for $i \in [1,n], i \neq \pi$.
  - Computes $c_\pi = f(\pi)$, $u_{\pi 1} = (psk_{ID\pi})^{r_{\pi 1} - c_\pi} \bmod N$, $u_{\pi 2} = r_{\pi 2} - x_{ID\pi} c_\pi$.
  - Outputs the ring signature on message m as $\sigma = (m, f, (u_{i1}, u_{i2})i \in [1,n])$.
- **Verify:** Given a CL-RS $\sigma = (m, f, (u_{i1}, u_{i2})i \in [1,n])$ on message m, the verifier executes as follows:
  - Checks if f is a polynomial over $GF(2^k)$ with degree $n-1$.
  - Computes $c_i = f(i), R_{i1} = u_{i1}^e H_1(ID_i)^{c_i} \bmod N, R_{i2} = H_1(ID_i)^{u_{i2}} upk_{IDi}^{c_i} \bmod N$ for $i \in [1,n]$.
  - Accepts the signature if and only if the following equation holds $f(0) = H_2(m||L_{ID}||L_{upk}||(R_{i1}, R_{i2})i \in [1,n])$.

3.2 Attack of Qin *et al.*'s CL-RS Scheme by TypeI Adversary

Qin *et al.* proved their scheme is secure against the two types of adversaries in CL-RS scheme. However, we found that the Type I adversary can forge the ring signature. $\mathcal{A}_1$ forges ID_π's signature as follows:

1) $r_{\pi 1}, r_{\pi 2}, R_{\pi 1}, R_{\pi 2}, \{c_i, u_{i1}, u_{i2}, R_{i1}, R_{i2}\}_{(i \in [1,n], i \neq \pi)}, f$ are generated as Qin *et al.*'s scheme.

2) $\mathcal{A}_1$ computes $c_\pi = f(\pi)$. If $r_{\pi 1} - c_\pi$ is not divided by e, $\mathcal{A}_1$ operates the step **Ring-Sign** of Qin *et al.*'s scheme.

3) If $r_{\pi 1} - c_\pi = eh$, $\mathcal{A}_1$ sets $u_{\pi 1} = H(ID_\pi)^h \bmod N$, $u_{\pi 2} = r_{\pi 2} - x'_{ID\pi} c_\pi$, $\sigma = (m, f, (u_{i1}, u_{i2})i \in [1,n])$ as the forged signature.

The forged signature can pass the verification:

$$\begin{aligned}
R_{\pi 1} &= u_{\pi 1}^e H_1(ID_\pi)^{c_\pi} \\
&= H_1(ID_\pi)^{eh} H_1(ID_\pi)^{c_\pi} \\
&= H_1(ID_\pi)^{r_{\pi 1} - c_\pi} H_1(ID_\pi)^{c_\pi} \\
&= H_1(ID_\pi)^{r_{\pi 1}} \bmod N \\
R_{\pi 2} &= H_1(ID_\pi)^{u_{\pi 2}} (upk'_{ID\pi})^{c_\pi} \\
&= H_1(ID_\pi)^{r_{\pi 2} - x'_{ID\pi} c_\pi} H_1(ID_\pi)^{x'_{ID\pi} c_\pi} \\
&= H_1(ID_\pi)^{r_{\pi 2}} \bmod N \\
f(0) &= H_2(m||L_{ID}||L_{upk}||(R_{i1}, R_{i2})i \in [1,n])
\end{aligned}$$

For the reason that $r_{\pi 1}$ is a random number, c_π is generated by polynomial f decided by random numbers $c_i (i \in [1,n], i \neq \pi)$ and hash function H_2 which could treated as a random number. The probability that $r_{\pi 1} - c_\pi$ dividing by e holds is $1/e$ which is no negligible. In conclusion, the Type I adversary can forge the CL-RS in a non-negligible probability.

4 Conclusion

Certificateless public key cryptography could eliminate the use of certificates in Public Key Infrastructure and solve the key-escrow problem in ID-based public key cryptography. Certificateless ring signature schemes can provide anonymous authentication for ad hoc networks. Recently, Qin et al. proposed a RSA-based CL-RS scheme which was proved unforgeable in random oracle model. However, we found that the scheme was not secure against the Type I adversary. In the future, we will design a more efficient CL-RS scheme without bilinear pairing. The novel scheme should be unforgeable in random oracle model.

Acknowledgements

This work is supposed by the National Nature Science Fundation of China (No.61173188, 61572001), China Postdoctoral Science Foundation (No.2015M570545), Anhui Provincial Natural Science Foundation (No.201508085QF132), the Educational Commission of Anhui Province, China(KJ2015A326), and the Open Project of Co-Innovation Center for Information Supply & Assurance Technology, Anhui University (No.ADXXBZ2014-9). All authors thank referees for their valuable suggestions.

References

[1] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," in *Advances in Cryptology (Asiacrypt'02)*, pp. 415–432, Springer, 2002.

[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology (Asiacrypt'03)*, pp. 452–473, Springer, 2003.

[3] M. H. Au, J. K. Liu, T. H. Yuen, and D. S. Wong, "ID-based ring signature scheme secure in the standard model," in *Advances in Information and Computer Security*, pp. 1–16, Springer, 2006.

[4] S. Chang, D. S. Wong, Y. Mu, and Z. Zhang, "Certificateless threshold ring signature," *Information Sciences*, vol. 179, no. 20, pp. 3685–3696, 2009.

[5] S. S. Chow and W. Yap, "Certificateless ring signatures," *IACR Cryptology ePrint Archive*, vol. 2007, pp. 236, 2007.

[6] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, DOI: 10.1109/JSYST.2015.2428620, 2015.

[7] J. Herranz, "Identity-based ring signatures from rsa," *Theoretical Computer Science*, vol. 389, no. 1, pp. 100–117, 2007.

[8] S. Horng, S. Tzeng, P. Huang, and et al., "An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Information Sciences*, vol. 317, no. 1, pp. 48–66, 2015.

[9] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Topics in Cryptology (CT-RSA'05)*, pp. 275–292, Springer, 2005.

[10] Z. Qin, H. Xiong, G. Zhu, and Z. Chen, "Certificate-free ad hoc anonymous authentication," *Information Sciences*, vol. 268, pp. 447–457, 2014.

[11] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology (Asiacrypt'01)*, pp. 552–565, Springer, 2001.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, pp. 47–53, Springer, 1985.

[13] M. Tian, W. Yang, and L. Huang, "Cryptanalysis and improvement of a certificateless multi-proxy signature scheme," *Fundamenta Informaticae*, vol. 129, no. 4, pp. 365–375, 2014.

[14] P. P. Tsang, M. H. Au, J. K. Liu, W. Susilo, and D. S. Wong, "A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity," in *Proceedings of 4th International Conference on Provable Security (ProvSec'10)*, LNCS 6402, pp. 166–183, Springer, 2010.

[15] D. H. Yum and P. J. Lee, "Generic construction of certificateless signature," in *Information Security and Privacy*, pp. 200–211, Springer, 2004.

[16] F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings," in *Advances in Cryptology (Asiacrypt'02)*, pp. 533–547, Springer, 2002.

[17] J. Zhang and J. Mao, "An efficient rsa-based certificateless signature scheme," *Journal of Systems and Software*, vol. 85, no. 3, pp. 638–642, 2012.

[18] L. Zhang, F. Zhang, and W. Wu "A provably secure ring signature scheme in certificateless cryptography," in *Provable Security*, pp. 103–121, Springer, 2007.

Yan Xu received her Ph.D degrees from University of Science and Technology of China in 2015. She is a lecturer at Anhui University. Her research interests include digital signature, cryptography.

Liusheng Huang is currently a professor and Ph.D supervisor in School of Computer Science and Technology at University of Science and Technology of China. His research interests include information security, wireless sensor network and distributed computing. He is author or coauthor of more than 100 research papers and six books.

Miaomiao Tian received his Ph.D degrees from University of Science and Technology of China in 2014. He is a postdoctor at University of Science and Technology of China. His research interests include information security, cryptography, digital signature.

Hong Zhong received her Ph.D degrees from University of Science and Technology of China in 2005. She is a professor and Ph.D supervisor in School of Computer Science and Technology at Anhui University. Her research interests include information security, cryptography.

Notes on "An Anonymous Multi-server Authenticated Key Agreement Scheme Based on Trust Computing Using Smart Card and Biometrics"

Yanjun Liu[1,2], Chin-Chen Chang[2,3] and Chin-Yu Sun[4]
(Corresponding author: Ching-Chun Chang)
School of Computer Science and Technology, Anhui University[1]
No. 111 Jiulong Rd., Hefei 230601, China
Department of Computer Science and Information Engineering, Asia University[2]
No. 500, Lioufeng Rd., Wufeng, Taichung 413, Taiwan
Department of Information Engineering and Computer Science, Feng Chia University[3]
No. 100 Wenhwa Rd., Seatwen, Taichung 407, Taiwan
Department of Computer Science, National Tsing-Hua University[4]
No. 101, Section 2, Kuang-Fu Road, Hsinchu, Hsinchu 30013, Taiwan
(Email: alan3c@gmail.com)
(Received Nov. 27, 2013; revised and accepted May 5 & July 25, 2014)

Abstract

Nowadays, multi-server remote user authentication schemes have been studied extensively in the literature. Recently, Chuang and Chen proposed a multi-server authentication scheme based on trust computing using smart cards and biometrics. Their scheme is more efficient and can achieve more security requirements than other related schemes. However, we found that Chuang and Chen's scheme can disclose private information shared between a legal user and an authorized server to another server. Moreover, loss of smart card attacks can be amounted and user anonymity cannot be achieved.

Keywords: Anonymity, authentication, disclosure of privacy, loss of smart card attack, multi-server

1 Introduction

Along with the rapid development of wireless communication technologies, more and more people can acquire different types of Internet service through their mobile devices effortlessly. Therefore, how to verify the validity of remote login users before they access the services has become a significant security problem in wireless networks. A remote user authentication scheme based on smart card [2, 3] and password is the most extensively used mechanism to solve the aforementioned security problem due to its simplicity and high efficiency.

Nowadays, the multi-server environment [1, 4, 5] has attracted increasing popularity such that a user may acquire services provided by multiple servers simultaneously. As a result, a practical remote user authentication scheme must take the multi-server environment into account to satisfy the requirement of single registration, that is, any user only needs to register at the registration center (RC) once without registering to each server. This can simplify the registration procedure and diminish computational burden of the RC's.

However, one of the shortcomings of conventional multi-server remote user authentication schemes is that if both the user's smart card and password are stolen, authentication schemes may be susceptible to some malicious attacks. To enhance the degree of security, Chuang and Chen [1] proposed a multi-server authentication scheme based on trust computing that integrates the user's unique biometrics (e.g., fingerprints and irises) with smart card and password. Their scheme is more efficient and can achieve more security requirements than other related schemes. Unfortunately, we found that Chuang and Chen's scheme suffered from some security weaknesses. More specifically, their scheme can disclose private information shared between a legal user and an authorized server to another server. It cannot withstand loss of smart card attacks and is not able to ensure user anonymity either. These security weaknesses will be demonstrated and analyzed in detail in the following section.

2 Comment on Chuang and Chen's Scheme

In this section, we first briefly review the multi-server authentication scheme proposed by Chuang and Chen [1], and then discuss its security weaknesses.

2.1 Review of Chuang and Chen's Scheme

Chuang and Chen's scheme [1] contains three types of entities, i.e., the user, the registration center (RC) and multiple servers. These entities perform three phases: 1) the registration phase; 2) the login and authentication phase; and 3) the password change phase. In the registration phase, servers and users must register at the RC respectively. Based on the concept of trust computing, all authorized servers constitute an alliance and trust each other. Each authorized server must register at the RC and share a common secret key PSK with the RC before providing services to users. Furthermore, Chuang and Chen's scheme assumes that the key PSK is impossible to be compromised and it will be used in the login and authentication phase later. On the other hand, each user only needs to register at the RC once without repeating registration to each server. After registration, the login and authentication phase is executed to achieve mutual authentication between the user and the server. In addition, users can select and update their passwords easily without depending on the RC. In the following, we give the detailed description of the user registration phase and the login and authentication phase in Chuang and Chen's scheme, and the notations used throughout the scheme are listed in Table 1.

Table 1: Notations list in Chuang and Chen's scheme

U_i	The user i
RC	The registration center
S_j	The authorized server j
ID_i	The identity of user U_i
PW_i	The password of user U_i
BIO_i	The biometrics information of user U_i
x	A secret value of RC
SID_j	The identity of authorized server S_j
N_i	A random number
$h(\cdot)$	A collision-free one-way hash function

2.1.1 The User Registration Phase

1) User U_i sends ID_i and $h(PW_i \oplus BIO_i)$ to the RC through a secure channel.

2) The RC computes $A_i = h(ID_i \parallel x)$, $B_i = h(A_i)$, $C_i = h(PW_i \oplus BIO_i) \oplus B_i$, and $D_i = PSK \oplus A_i$.

3) The RC stores the parameters $\{ID_i, B_i, C_i, D_i, h(\cdot)\}$ on a new smart card and issues the smart card to user U_i over a secure channel.

2.1.2 The Login and Authentication Phase

1) User U_i inserts his/her smart card into a card reader and then inputs his/her ID_i and PW_i and scans his/her BIO_i at the sensor.

2) The smart card checks ID_i and then examines whether $h(PW_i \oplus BIO_i) \oplus C_i$ is equal to B_i or not. If the equation holds, the smart card generates a nonce N_1, and then computes $M_1 = h(B_i) \oplus N_1$, $AID_i = h(N_1) \oplus ID_i$, and $M_2 = h(N_1 \parallel AID_i \parallel D_i)$.

3) The smart card sends the authentication message $\{AID_i, M_1, M_2, D_i\}$ to server S_j.

4) Server S_j retrieves $A_i = D_i \oplus PSK$ and $N_1 = M_1 \oplus h^2(A_i)$. Then,$S_j$ computes and checks $h(N_1 \parallel AID_i \parallel D_i)? = M_2$. If it holds, the phase continues; otherwise,S_j terminates the phase. Next, S_j generates a nonce N_2 and constructs the session key $SK_{ij} = h(N_1 \parallel N_2)$. After that, S_j computes $M_3 = N_2 \oplus h^2(N_1)$ and $M_4 = h(SID_j \parallel N_2)$.

5) Server S_j sends the authentication reply message $\{SID_j, M_3, M_4\}$ to the smart card.

6) The smart card retrieves $N_2 = M_3 \oplus h^2(N_1)$ and checks whether $h(SID_j \parallel N_2)$ is equal to M_4 or not. If it holds, the smart card can generate the session key $SK_{ij} = h(N_1 \parallel N_2)$ and $M_5 = SK_{ij} \oplus h(N_2)$.

7) The smart card sends M_5 to server S_j.

8) Server S_j retrieves $h(N_2) = M_5 \oplus SK_{ij}$ and checks the validity of this value.

If the authentication is passed, the server and the user can mutually authenticate each other and establish a shared session key SK_{ij} for the subsequent secret communication. In Subsections 2.2 - 2.4, we will show the security weaknesses of this authentication scheme.

2.2 Disclosure of Privacy

In the multi-server environment, a user does not need to register to each server but only registers to the RC once [4, 5]. Moreover, Chuang and Chen assumed that their multi-server authentication scheme are based on trust computing, which means all authorized servers can trust and work in close collaboration with each other. Although authorized servers can be considered as an alliance in Chuang and Chen's scheme, it does not imply that one authorized server has the privilege to access the private information shared between a user and another authorized server. Unfortunately, we have found that the session key shared between a legal user and an authorized server can be disclosed to another authorized server. Under the

assumption that there are three entities, i.e., user U_i and servers S_A and S_B, and SK_{iA} is the session key shared between U_i and S_A, we demonstrate how S_B obtains SK_{iA} without detection by the following steps.

1) Server S_B registers at the RC and shares a secret key PSK with the RC.
2) S_B intercepts the messages M_1^A, D_i^A, and M_3^A that are transmitted between user U_i and server S_A through the public channel in the authentication.
3) S_B retrieves $A_i^A = D_i^A \oplus PSK$ and then uses M_1^A and A_i^A to compute $N_1^A = M_1^A \oplus h^2(A_i^A)$.
4) S_B uses M_3^A and N_1^A to obtain $N_2^A = M_3^A \oplus h^2(N_1^A)$.
5) With N_1^A and N_2^A in hand, S_B can immediately extract the session key $SK_{iA} = h(N_1^A \parallel N_2^A)$ shared between user U_i and server S_A.

2.3 Loss of Smart Card Attack

Here, we explain why Chuang and Chen's scheme is unable to withstand loss of smart card attacks. Assuming that an attacker Eve has stolen user U_i's smart card, Eve can extract all the secret information, i.e., ID_i, B_i, C_i, D_i, and $h(\cdot)$ preserved in the smart card and successfully launches the loss of smart card attack without knowing U_i's password PW_i and biometrics BIO_i as follows.

1) Without U_i's correct parameter N_1, Eve must choose a random number N_1^* and generates $M_1^* = h(B_i) \oplus N_1^*$, $AID_i^* = h(N_1^*) \oplus ID_i$, and $M_2^* = h(N_1^* \parallel AID_i^* \parallel D_i)$ by himself/herself.
2) Eve impersonates U_i to send $\{AID_i^*, M_1^*, M_2^*, D_i\}$ to server S_j.
3) S_j retrieves $A_i = D_i \oplus PSK$ and $N_1^* = M_1^* \oplus h^2(A_i)$. Then, S_j checks whether $h(N_1^* \parallel AID_i^* \parallel D_i)$ is equal to M_2^*. If it holds, S_j continues the procedure.
4) S_j generates a nonce N_2 and constructs the session key $SK_{Ej} = h(N_1^* \parallel N_2)$. Afterwards, S_j computes $M_3^* = N_2 \oplus h^2(N_1^*)$ and $M_4 = h(SID_j \parallel N_2)$.
5) S_j sends $\{SID_j, M_3^*, M_4\}$ to Eve.
6) Eve retrieves $N_2 = M_3^* \oplus h^2(N_1^*)$ and checks whether $h(SID_j \parallel N_2)$ is equal to M_4. If it holds, Eve computes the session key $SK_{Ej} = h(N_1^* \parallel N_2)$ and $M_5^* = SK_{Ej} \oplus h(N_2)$.
7) Eve sends M_5^* to S_j.
8) S_j retrieves $h(N_2) = M_5^* \oplus SK_{Ej}$ and checks the validity of this value.

Based on the above analysis, it indicates that when the smart card is stolen, the server can be convinced that attacker Eve is a legal user and they will establish a common session key. Therefore, loss of smart card attacks can be amounted in Chuang and Chen's scheme.

2.4 User Anonymity

Chuang and Chen claimed that their scheme can ensure the user anonymity such that an attacker has no way to obtain the original identity of a user. This is because the user's identity is concealed in AID_i as $AID_i = h(N_1) \oplus ID_i$ and the random number N_1 selected by user U_i is not revealed. However, the following scenario shows that attacker Eve can determine the original identity of a user.

1) Attacker Eve intercepts the messages AID_i, M_2, and D_i.
2) Since $M_2 = h(N_1 \parallel AID_i \parallel D_i)$, Eve can easily find out the correct value of N_1 via M_2, AID_i, and D_i by launching an off-line guessing attack.
3) Eve computes $ID_i = AID_i \oplus h(N_1)$.

Therefore, we can conclude that Chuang and Chen's scheme cannot achieve user anonymity.

2.5 Conclusions

In this paper, we pointed out the security weaknesses in the multi-server authentication scheme based on trust computing proposed by Chuang and Chen. Although their scheme combines the user's biometrics with smart card and password to enhance the security, it still suffers from three security problems, i.e., 1) the disclosure of the session key shared between a legal user and an authorized server; 2) it cannot withstand loss of smart card attacks; and 3) it cannot guarantee user anonymity.

References

[1] 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.
[2] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Communications*, vol. 34, no. 3, pp. 367–374, 2011.
[3] C. T. Li, and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for wireless communications," *Mathematical and Computer Modelling*,vol. 55, no. 1, pp. 35–44, 2012.
[4] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.
[5] W. J. Tsaur, J. H. Li, and W. B. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.

Yanjun Liu received her B.S. degree in 2005, in School of Computer Science and Technology from Anhui University, Hefei, China. She received her Ph.D. degree in 2010, in School of Computer Science and Technology from University of Science and Technology of China, Hefei, China. She is currently serving in Anhui University. Meanwhile, she is a postdoctor at Asia University, Taichung, Taiwan. Her current research interests include information security and computer cryptography.

Chin-Yu Sun received the MS degree in Department of Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan in 2013. He is currently pursuing his Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan. He current research interests include information security, cryptography, wireless communications, mobile communications, and cloud computing.

Chin-Chen Chang received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Taiwan. He was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the College of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of the R.O.C. Since February 2005, he has been a Chair Professor of Feng Chia University. He is currently a Fellow of IEEE and a Fellow of IEE, UK. He also published several hundred papers in Information Sciences. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

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.